

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií

DIPLOMOVÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

TVORBA A DEMONSTRACE 3D MODELŮ PRO VR
CREATION AND DEMONSTRATION OF ASSETS FOR VR APPLICATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MAREK ZOUHAR

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. TOMÁŠ MILET

BRNO 2017

Abstrakt

Tato práce pojednává o konceptu virtuální reality, její historii, současných možnostech a dostupných zařízeních a technologiích virtuální reality a způsobech tvorby prvků, zejména modelů, textur a animací, které aplikace ve virtuální realitě využívá. Praktická část této práce se zabývá návrhem a tvorbou trojrozměrných modelů, textur, animací a prostředí pro využití v interaktivní aplikaci ve virtuální realitě a také návrhem a tvorbou takové aplikace, která jejich využití demonstruje.

Abstract

This thesis deals with the concept of virtual reality, its history, present-day possibilities and available devices and technologies for virtual reality and ways of creating assets such as models, textures and animations for virtual reality applications. The practical part of this work deals with design and creation of three-dimensional models, textures, animations and environment for use in interactive application in virtual reality and also with design and creation of such application to demonstrate their use.

Klíčová slova

Virtuální realita, počítačová grafika, 3D modely, polygonální modely, textury, normálová mapa, bump mapa, animace, Unity, Blender, HTC Vive

Keywords

Virtual reality, computer graphics, 3D models, polygonal models, textures, normal map, bump map, animation, Unity, Blender, HTC Vive

Citace

Zouhar Marek: Tvorba a demonstrace 3D modelů pro VR, diplomová práce, Brno, FIT VUT v Brně, 2017

Tvorba a demonstrace 3D modelů pro VR

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Tomáše Mileta. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Marek Zouhar
11. ledna 2017

Poděkování

Na tomto místě bych rád poděkoval panu Ing. Tomáši Miletovi za vedení kvalifikační práce a své rodině a přátelům za podporu.

© Marek Zouhar, 2017

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Virtuální realita	4
2.1	Historie	4
2.2	Současná technologie.....	7
2.2.1	HMD – Head Mounted Display	8
2.2.2	Alternativní zobrazení	12
2.2.3	Ovladače.....	13
3	Modely.....	19
3.1	Reprezentace.....	19
3.2	Tvorba.....	20
3.2.1	Modelovací nástroje.....	20
4	Návrh aplikace	22
4.1	Uživatelské rozhraní	22
4.1.1	Tradiční prvky pro VR	22
4.1.2	Přístup k hlavním objektům	23
4.2	Modely	24
4.2.1	Scéna	24
4.2.2	Střelné zbraně.....	24
4.2.3	Meč a vrhací předměty	25
4.2.4	Další objekty	25
5	Implementace.....	26
5.1	Implementace chování	26
5.1.1	Uživatel	26
5.1.2	Předměty	27
5.1.3	Scéna	29
5.2	Tvorba modelů, textur a animací	31
5.2.1	Modely	31
5.2.2	Textury	32
5.2.3	Animace	34
6	Testování.....	36
7	Závěr	38
	Literatura.....	39
	Obsah DVD.....	43
	Manuál	44
	Příloha 1	45

Seznam obrázků

Obrázek 1: Sensorama	5
Obrázek 2: DataGlove (vlevo), PowerGlove (uprostřed) a Provision 100 (vpravo).....	6
Obrázek 3: Oculus Rift (vlevo) a HTC Vive (vpravo).....	8
Obrázek 4: PlayStationVR (vlevo) a StarVR (vpravo).....	10
Obrázek 5: Google Cardboard (vlevo) a Samsung GearVR (vpravo)	11
Obrázek 6: Cave Automatic Virtual Environment.....	12
Obrázek 7: Ovladač HTC Vive (vlevo) a Oculus Touch (vpravo)	14
Obrázek 8: GROPE III (vlevo) a Phantom Premium (vpravo).....	15
Obrázek 9: Senso Glove (vlevo) a Perception Neuron (vpravo).....	16
Obrázek 10: Tesla Suit (vlevo), Virtuix Omni (uprostřed) a InfinaDeck (vpravo).....	18
Obrázek 11: Hlavní vstupy ovladače Vive (vlevo) a vytvořené kruhové menu (vpravo).....	23
Obrázek 12: Ilustrace zobrazení plného a poloprázdného zásobníku	24
Obrázek 13: Ilustrace UV mapování textur	32
Obrázek 14: Příklad různých typů vytvořených textur (raketomet, stěna a podlaha).....	33
Obrázek 15: Ilustrace průběhu tvorby animací	34

1 Úvod

Virtuální realita (VR) se v posledních letech dostala do povědomí široké veřejnosti. V následujících kapitolách ale ukážeme, že se ve skutečnosti jedná již o poměrně dlouho zavedený koncept. Kromě historie prozkoumáme vstupní i výstupní zařízení pro užití s virtuální realitou, které jsou aktuálně dostupné nebo ve vývoji. Asi největší pozornost z nich bude věnována náhlavním displejům, které jsou dnes s VR neoddělitelně spjaty, ale ukážeme, že lze vytvořit obdobně uvěřitelné virtuální prostředí i bez nich. Dalším tématem budou ovladače vcelku tradiční ale i futuristické jako jsou rukavice s dotykovou odezvou, celotělové obleky nebo všesměrové běžící pásy.

Virtuální realita silně prohlubuje uplatnění 3D grafiky v oblastech jako je zábavní průmysl, ve výzkumu, ve zdravotnictví, výrobě a spoustě dalších odvětví a tak se v další kapitole budeme věnovat stručnému shrnutí možností reprezentace 3D modelů a dostupným programům pro jejich tvorbu.

Nebot' se praktická část této práce zabývá tvorbou trojrozměrných modelů, jejich textur a animací a tvorbou aplikace ve virtuální realitě, která tyto prvky umožní demonstrovat v praxi, budeme se také podrobně věnovat návrhu této aplikace i modelů které pro ni byly vytvořeny.

V další kapitole probereme implementaci zmíněné aplikace a nastíníme problematiku tvorby modelů a definice materiálů definujících jejich povrch pomocí vytvořených textur.

Text práce zakončíme několika poznatky získanými při testování aplikace a v závěru zhodnotíme dosažené výsledky.

V rámci semestrálního projektu byly sepsány kapitoly 2 a 3. Co se týče praktické části této práce, semestrální projekt pokrývá tvorbu přibližně 50 % modelů, 30 % textur, 30 % animací a experimenty s enginem Unity a integrací vytvořených modelů do virtuální reality a interakce v ní.

2 Virtuální realita

Než budeme pokračovat, je důležité definovat, co přesně máme chápat pojmem *virtuální realita* (VR) a co už virtuální realita není. Oddělme nejprve od VR takzvanou *rozšířenou realitu* (*Augmented Reality* – AR). Ta dostala svůj název proto, že rozšiřuje skutečný svět tím virtuálním. V dnešní době je to asi nejsnáze možné demonstrovat na speciálních brýlích od společností Google (Glass) nebo Microsoft (HoloLens) [1], skrz které uživatel vidí jako obvykle, ale do oka jsou mu navíc promítány další informace, jako například instrukce GPS navigace. Podobně jako AR, virtuální realita rovněž ve striktním slova smyslu není oklamáním smyslů člověka, aby měl pocit, že se nachází na jiném skutečném místě (telepresence), což bývá poměrně často užíváno v telerobotice pro ovládání strojů na dálku [2].

V minulosti se také vyskytly pokusy definovat virtuální realitu pomocí technologií, které využívá. Je však poměrně zřejmé, že taková definice rovněž není příliš vhodná, vezmeme-li v úvahu, že ani náhlavní displej, který je dnes téměř synonymem pro virtuální realitu, je možné nahradit CAVE systémem [3], který využívá projektory promítající na stěny místnosti trojrozměrný polarizovaný obraz, který se mění v závislosti na poloze uživatele. Ostatně celá podkapitola je níže věnována možnostem, jaké má virtuální realita i z pohledu výstupních zařízení.

Virtuální realitu je vhodné definovat z pohledu funkčnosti. Jak uvádí Burdea a Coiffet [2], virtuální realita je simulace, ve které je užita počítačová grafika k vytvoření světa podobného skutečnosti. Tento svět navíc nemá být statický, ale má okamžitě reagovat na vstup uživatele. Autoři uvádí, že virtuální realita je definována třemi I: Immersion, Interaction a Imagination, tedy pohroužení uživatele, vzájemné ovlivňování a představitost, která má být užita k tvorbě virtuálních světů a interakci v nich.

2.1 Historie

Ačkoli, jak bylo zmíněno v úvodu této práce, se virtuální realita dostala do povědomí veřejnosti teprve nedávno, její počátky sahají překvapivě hluboko do minulosti. Ve skutečnosti už dokonce jeden takový svůj menší „boom“ zažila koncem osmdesátých let minulého století, ale nadšení okolo ní nakonec možná kvůli zatím nedostatečně vyvinuté technologii (a tehdejším přílišným očekáváním veřejnosti, které se daří vyplnit teprve dnes) poněkud pohaslo.

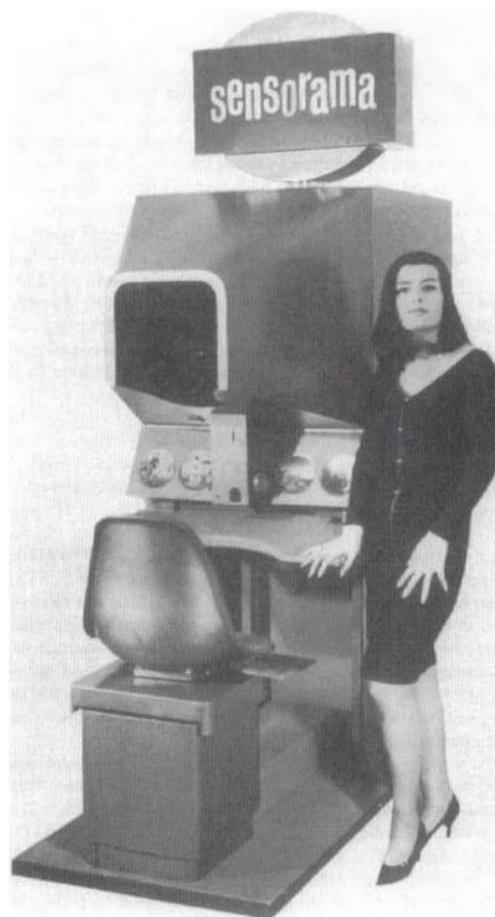
Pokud ovšem vykročíme směrem od naší stanovené definice, abychom našli nejstarší náznaky toho, co nakonec vyústilo v dnešní virtuální realitu, můžeme se dostat až do novověku a jeho realistickým obrazům. Panoramatické malby a dokonce třistašedesátistupňové fresky devatenáctého století měly rozhodně za úkol vtáhnout pozorovatele do děje či scénérie, jež zobrazovaly [4].

Počátky té pravé virtuální reality nalezneme ve dvacátém století, ať už mluvíme o roce 1929 a leteckém simulátoru Edwina Alberta Linka (*Link trainer* či *Blue box*) [4], který byl používán při tréninku pilotů, nebo o padesátých letech, kdy Morton Heilig představil světu svůj vynález *Sensorama* (Obrázek 1) [2] [4] [5]. Heilig svou vizi publikoval v roce 1955 jako „Kino budoucnosti“ a v roce 1962 patentoval a postavil její prototyp [6]. Jedná se o stroj s vibračním sedadlem, stereoskopickým obrazem, stereo zvukem a dokonce s větráky simulujícími vítr a zařízením reprodukcí pachy. Toto zařízení dokázalo působivě navodit pocit například jízdy na motorce Brooklynem.

Heilig byl skutečně plodným vynálezcem na poli virtuální reality. Opět jemu totiž můžeme připisat vynález náhlavního displeje se stereo zvukem patentovaným už v roce 1960 [7] a „divadlo zážitků“ (Experience Theatre) patentované v roce 1969 [8]. Poslední jmenovaný vynález je verze

Sensoramy pro více diváků s velkým plátnem tvaru výseče koule s trojrozměrným obrazem, směrovým zvukem, vůněmi, větrem, změnami teploty a nakláněním sedadel.

V roce 1968 vytvořil Ivan Sutherland s pomocí svého žáka Boba Sproulla první náhlavní displej (Head-Mounted Display – HMD). Jak je zřejmé například už z patentu Mortona Heiliga, nejedná se o první HMD v principu, ale ve skutečném provedení. Jeho displeje byly velmi těžké CRT monitory, kvůli čemuž muselo být zařízení neseno robotickým ramenem umístěným na stropě nad uživatelem [2], čímž si zařízení získalo svůj název „Damoklův meč“ [5]. Virtuální prostředí bylo tehdy velice jednoduché – pokoje reprezentované formou drátěného modelu a první experiment byla pouze krychle levitující před uživatelem. Při práci na svém HMD si Sutherland uvědomil, že lze zobrazovat scény kompletně generované počítačem, založil se svým kolegou Davidem Evansem společnost Evans and Sutherland a věnovali se této problematice. Kolem roku 1973 dokázali zobrazit scénu o 200 až 400 polygonech dvacetkrát za vteřinu a docílit tak iluze animace [2]. Sutherland také již v roce 1965 předpověděl vývoj hmatové odezvy pro virtuální realitu a tato myšlenka byla převedena do skutečnosti Frederickem Brooksem roku 1971 [2] [5]. Ten dokázal se svými kolegy systémem nazvaným GROPE simulovat

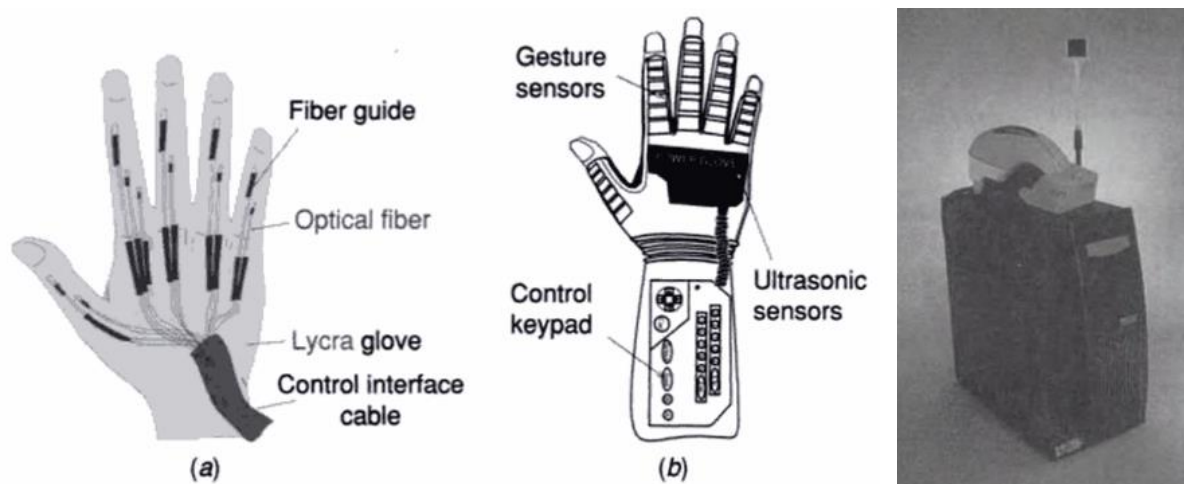


Obrázek 1: Sensorama

nejprve dvourozměrné a později (s využitím robotického ramene používaného pro práci s radioaktivním materiálem) trojrozměrné silové působení způsobené kolizí objektů. Uživatel ramenem volně manipuluje v prostoru, avšak pokud software vyhodnotí, že se systém ocitl ve virtuálním prostředí na hranici objektu, začne klást odpor. Výsledkem je pocit, že v cestě stojí skutečný předmět.

K vývoji technologií virtuální reality přispěla svým dílem i armáda. Letecké simulátory byly totiž vyráběny vždy pro dané modely letadel, ovšem jakmile se stal daný model letounu zastaralý, bylo nutné pořídit nový velmi drahý simulátor. Snahy o snížení výdajů tedy vedly k pokusům o přesun této části do softwaru simulátoru. Během sedmdesátých a osmdesátých let bylo toto pole intenzivně studováno, avšak většina výzkumů byla utajována. To se změnilo, když přestaly být některé výzkumy financovány, a vědci, kteří na nich pracovali, se přesunuli do civilního sektoru [2].

Kvůli obtížně reprodukovatelným podmínkám v kosmu měla zájem na vývoji virtuální reality také NASA. V roce 1984 vyvinula pro účely tréninku astronautů pokročilý simulátor VIVED (Virtual Visual Environment Display) [5] se stereoskopickým monochromatickým náhlavním displejem. V prototypu zařízení byly použity LCD displeje jednoduše vymontované z televizoru Samsung Watchman a před ně byly umístěny čočky pro pohodlné zaostření. Jako pomocnou výpočetní jednotku VIVED využíval stroj již zmíněné společnosti Evans and Sutherland [2]. Projekt se vyvíjel dále a s příchodem Scotta Fishera a Elizabeth Wenzelové do něj do roku 1988 přibyla nová snímací rukavice a podpora až čtyř virtuálních prostorových zdrojů zvuku. Převodem softwaru na výkonnější stroj, který umožnil místo drátěného modelu zobrazovat vykreslené plochy, se stal původní projekt projektem novým – VIEW (Virtual Interface Environment Workstation).



Obrázek 2: DataGlove (vlevo), PowerGlove (uprostřed) a Provision 100 (vpravo)

Důležitým milníkem je též prodej snímací rukavice DataGlove od společnosti VPL v roce 1985 (Obrázek 2) [5] [2]. Rukavice měla na každém prstě umístěné optické vlákno a podle množství jím průchozího světla určovala míru ohybu jednotlivých prstů. Tato technologie představovala velký krok kupředu v přirozeném způsobu komunikace s počítačem, ale technologie nebyla bez chyb. Kromě vysoké ceny (která se tehdy pohybovala v tisících dolarů) byly problémy s jednotnou velikostí rukavice a absence dotykové zpětné vazby. Krátce po přivedení DataGlove na trh představila společnost Nintendo svou verzi rukavice – mnohem levnější PowerGlove (Obrázek 2) [2]. Rukavice využívala ultrazvukových senzorů pro určení polohy vůči obrazovce a omezeně snímala i ohyb prstů. Ovladač se sice zpočátku prodával poměrně dobře, ale nedostatek her, které by jej využívaly, nakonec vedl k zastavení jeho produkce.

Opět společnost VPL na trh přinesla roku 1988 komerční náhlavní displej EyePhones [5] [2]. HMD pro zobrazení stereoskopického obrazu využíval dva LCD displeje, ale jejich rozlišení bylo bohužel pouhých 360×240 pixelů, takže se scéna jevila jako rozmazaná. Dalšími nevýhodami byla velmi vysoká váha (2,4 kg) a ještě vyšší cena kolem 11 000 dolarů.

Vývoj pokračoval směrem k vytvoření kompletní jednotky pro běžného uživatele, která má všechny potřebné části a periferie, na kterou bude možné vytvářet jednotné aplikace. První pokus učinila nepřekvapivě společnost VPL, která vytvořila RB2 Model 2 obsahující mimo jiné periferie HMD EyePhones, VPL DataGlove Model 2 a jednotku pro čtení pozice HMD. Dalším krokem bylo zmenšení zařízení a umístění všech částí do kompaktní skříně. Tímto zařízením byla roku 1991 ve Velké Británii komerční VR stanice Vision společnosti Division Ltd., která byla brzy vylepšena na výkonnější model Provision 100 (Obrázek 2). Její helma pro srovnání se současnou technologií vážila 2 kg a její rozlišení bylo stále pouhých 360×240 pixelů. Technologický pokrok však nezastavitelně uháněl kupředu mílovými skoky, a tak se do roku 1997 rozlišení HMD LCD displejů zdvojnásobilo. O rok později Sony přivedlo na trh brýle Glasstron s rozlišením 800×600 pixelů a vahou pouhých 310 gramů. Krátce nato společnost Kaiser Electro-Optics představila náhlavní LCD displej o rozlišení 1024×768 , které bylo jinak dostupné pouze na mnohem dražších a těžších HMD s CRT displeji nebo na barevných monitorech desktopů [2]. Vyšší rozlišení znamenalo kvalitnější obraz a výkonnější stroje zase působivější scény, čímž se pomalu dostáváme do současnosti.

2.2 Současná technologie

Vývoj technologie virtuální reality, jak jsme právě viděli, již ušel dlouhou cestu. Věnovali se mu desítky let výzkumníci po celém světě, a tak se nemůžeme divit pokroku, kterého se podařilo dosáhnout. Jak již bylo zmíněno v úvodu této kapitoly, náhlavní displeje jsou dnes v povědomí většiny lidí prakticky synonymem virtuální reality, avšak ty jsou pouze zlomkem všech technologií, jež lze pro virtuální realitu využít. V této kapitole jmenujeme mnoho zástupců z řad vstupních a výstupních zařízení, které lze ve spojení s virtuální realitou využít. Nejedná se však v žádném případě o kompletní seznam všech výrobků, které jsou aktuálně vyvíjené či již na trhu, jde pouze o jistý přehled možností a srovnání dostupných (případně brzy dostupných) řešení. Ještě než se však budeme věnovat oné široké škále vstupních a výstupních zařízení, zmiňme technologické možnosti, které mohou tato zařízení využívat pro sledování své pozice a orientace, aby tyto informace využila pro tvorbu uvěřitelného virtuálního prostředí.

Pro virtuální realitu jsou vhodné sledovací zařízení (trackery), které mají 6 stupňů volnosti (6 DOF – Degree of Freedom), což pokrývá pozici a rotaci zařízení – obojí ve třech osách. Tyto trackery musí mít vhodné následující vlastnosti: [5]

- *Obnovovací frekvence* sledování určuje, kolikrát za vteřinu je změřena a odeslána informace o poloze a rotaci, což je klíčové pro snímání plynulého pohybu.
- *Latence*, neboli zpoždění, určuje dobu, která uběhne mezi skutečným pohybem uživatele a reakcí systému na něj.
- *Přesnost* měření udává nejmenší jednotku délky, či úhlu, kterou je zařízení schopné rozlišit.
- *Vzdálenost*, která vymezuje prostor, na kterém dokáže tracker, splnit výše zmíněné vlastnosti.

Během vývoje sledovacích zařízení byly použity následující přístupy pro sledování pohybu ovládacích prvků: [5]

Sledování pohybu může být uskutečněno například pomocí **magnetického pole**. Takové senzory jsou malé a lehké, není u nich vyžadováno, aby mezi vysílačem a senzorem byla nepřerušovaná linie pohledu (line of sight), mají nízkou latenci a dobrou obnovovací frekvenci a jsou dobře dostupné. Vzhledem k vlastnostem magnetického pole je však zřejmou nevýhodou tohoto řešení poměrně nízký dosah a velké rušení feromagnetickými prvky.

Jinou možností je využití **ultrazvuku**, kdy je změřena doba letu zvuku. Tyto systémy jsou rovněž malé, lehké a poměrně levné a navíc netrpí magnetickým rušením. Jsou ovšem nepřekvapivě rušeny hlukem či ozvěnou, nemají příliš vysoké obnovovací frekvence a vyžadují čistou linii pohledu.

Dnes nejpoužívanější volbou jsou **optická sledovací zařízení**. Jejich výhodami jsou vysoké obnovovací frekvence, možnost spravovat velkou plochu, dobrá přesnost a odolnost proti zmíněným zdrojům rušení. Nevýhodou pak je, že jsou dražší, než zmíněné alternativy, vyžadují čistou linii pohledu a v minulosti byl problém sledovat více objektů současně v jednom prostředí. Na těchto nedostatcích se však díky výhodám a rozšíření těchto systémů v posledních letech intenzivně pracovalo.

Pokud vyžadujeme velmi přesné sledování pohybu (například chirurgický zákrok), nemusí být však přesnost optických sledovacích systémů dostatečná. Tento nedostatek řeší **mechanické snímače**. Jejich dalšími výhodami jsou absolutní odolnost proti rušení, vysoké obnovovací frekvence a možná podpora silové zpětné vazby. Nevýhodami je naopak malá plocha, na které lze pracovat a omezená svoboda pohybu kvůli mechanickému spojení.



Obrázek 3: Oculus Rift (vlevo) a HTC Vive (vpravo)

2.2.1 HMD – Head Mounted Display

Náhlavní displej je nejčastější forma vizuálního výstupu dnešního VR. Obrovské množství společností v posledních letech soupeří o pozornost spotřebitelů, kteří mají na výběr z širokého spektra produktů. Všechny dnešní HMD mají do svého designu před displej integrovány optické čočky. Ty zajistí, že displej umístěný velmi blízko očím bude namapován na mnohem širší zorné pole (Field of View – FOV) a současně umožní očím uživatele jednodušší a pohodlnější zaostření na promítaný obraz. Použití čoček má ale své nevýhody. Hlavní dva problémy jsou barevná vada a distorze. Problém barevné chyby spočívá v tom, že index lomu pro různé vlnové délky není totožný a tak čočka nemůže zaostřit všechny barvy do jednoho ohniska. Distorze zase způsobuje, že aby uživatel viděl rovné čáry, je nutné je na displej vykreslit zakřivené dle parametrů čočky. Kvůli své charakteristice a fyzikální podstatě jsou oba problémy řešeny softwarově. První prototypy HMD, které měly nízká rozlišení, měly také nedokonalosti v tom, že na jejich displejích bylo možné vidět mezery mezi pixely.

2.2.1.1 Rift

Společnost Oculus vyvinula brýle Rift (Obrázek 3) [9], které jsou z velké části zodpovědné za zvýšení zájmu veřejnosti o virtuální realitu. První verze brýlí, která vyšla v polovině roku 2013, označovaná jako DK1 [10] (Development Kit 1) s cenou v přepočtu pod 10 000 korun byla určena především pro vývojáře aplikací na novou platformu, ale přesto si ji zakoupili i mnozí nadšenci a zájemci o technologii. Přestože se jednalo o nevýkonnější dosud sestavený HMD jeho omezený hardware nechával velký prostor pro budoucí zlepšení. Konkrétně rozlišení displeje pro každé oko bylo pouhých 640×480 pixelů [10]. Brýle snímaly svou rotaci ve všech třech osách a podle ní byl upravován vykreslený obraz. Fakt, že pozice hlavy uživatele ve virtuálním prostředí ale nezávisela na skutečné pozici, však v některých případech vedl k problémům s nevolností, se kterými vývojáři aplikací pro virtuální realitu musí dodnes bojovat.

V polovině příštího roku Oculus vydal druhou verzi brýlí pro vývojáře DK2. Její displej po nutném vylepšení dosáhl kombinovaného rozlišení 1920×1080 pixelů (960×1080 pro každé oko). Důležitý je též přechod z LCD na OLED displej, který je jasnější a má výhodnější vlastnosti, takže tolik nerozmazává pohyb, což se ukázalo jako nedostatek minulé verze a jeden z faktorů způsobující nevolnosti (**Virtual Reality Sickness**). Zajímavý trik druhé verze náhlavního displeje spočíval v tom, že namísto toho, aby byl vykreslený snímek zobrazený, dokud se nespočte ten nový, byly mezi snímky vkládány snímky černé, což má mít v lidském mozku za následek iluzi plynulejší animace. Diskutabilně největším rozdílem verze DK2 od DK1 ale je, že bylo implementováno sledování pozice brýlí

s využitím sensoru snímajícího infračervené LED diody na brýlích. To, spolu se snížením latence snímání natočení hlavy od minulé verze a výše zmíněnými vylepšeními, výrazně snížilo četnost problémů s nevolností uživatelů.

Dnes již je na trhu HD verze Riftu pro běžné uživatele [11] [9], která má rozlišení 2160×1200 (1080×1200 na jedno oko), zobrazovací frekvenci 90 Hz, zorné pole asi 110° a váhu jen 470 gramů. Tato verze brýlí byla v době tvorby této práce dostupná za cenu okolo 18 000 korun. Důležitým výstupem virtuální reality je samozřejmě i zvuk a Rift má tak pochopitelně integrován konektor pro sluchátka. Rift je dodáván s ovladačem herní konzole Xbox One, ovšem na trhu jsou nyní i nově vyvinuté ovladače Touch, kterým se budeme věnovat v další podkapitole. Rift je původně navržen pro užití vsedě před monitorem počítače, což lze naznat i z toho, že jeho výchozím ovladačem je ovladač konzole Xbox One a z toho, že starší verze brýlí nepodporovaly 360° sledování uživatele. Nová verze však tento nedostatek naštěstí odstranila. Navíc lze dokoupit další senzory, což umožní sledování uživatele v prostoru o velikosti pokoje.

2.2.1.2 Vive

Náhlaví displej vytvořený společností **HTC** ve spolupráci s **Valve** (která stojí za velmi populární platformou Steam) [12], nese název Vive (Obrázek 3) a je dnes diskutabilně největším konkurentem Riftu. To je způsobeno především tím, že pro sledování polohy a rotace brýlí Vive využívá dvou senzorů s technologií Lighthouse, což umožňuje již v základním balení brýlí přesné sledování polohy v oblasti až $4,6 \times 4,6$ metrů i s oběma dodávanými ovladači snímanými stejným způsobem, přičemž nezáleží na orientaci uživatele. Dle testů se snímaná pozice nehybného HMD pohybuje v kouli o průměru asi 0,3 milimetry, což je za hranicí vnímání člověkem [13].

Displej brýlí Vive technologicky odpovídá novému Riftu s rozlišením 2160×1200 (tedy 1080×1200 na jedno oko) a obnovovací frekvencí 90 Hz [11] [14]. Brýle mají dle specifikace přibližně stejné zorné pole 110° , to je však vhodné rozlišit na horizontální a vertikální. Dle praktických testů [15] má Vive asi 100° horizontální a 110° vertikální FOV oproti Riftu s 80° horizontálním a 90° vertikálním zorným polem, což je poměrně překvapivý rozdíl. Brýle mají samozřejmě podobně jako Rift integrovaný konektor pro sluchátka a jejich cena se nyní pohybuje okolo 26 000 korun, avšak tato cena je o tolik vyšší ve srovnání s Riftem zejména kvůli ovladačům, jež jsou součástí balení. Oboje zmíněné brýle vyžadují, aby byly připojeny k počítači. Tento připojovací kabel však bohužel představuje jisté nepohodlí a potenciálně nebezpečí, pokud se uživatel do drátů zamotá. Počítač, ke kterému jsou zařízení připojena, musí být pro vykreslení virtuálního prostředí dostatečně výkonný. Rift i Vive mají srovnatelné požadavky na jeho hardware, kde nejkritičtější komponenty přibližně odpovídají výkonu procesoru Intel Core i5-4590, grafické kartě NVIDIA GeForce GTX 1060 či AMD Radeon RX 480 a paměti RAM 8 GB (4GB pro Vive). Uvedme ještě pro úplnost hmotnost headsetu, která činí 555 gramů.

2.2.1.3 PlayStationVR

PSVR (Obrázek 4) je alternativa společnosti Sony pro její konzoli PlayStation 4. Dle specifikace [16] mají brýle opět OLED displej, tentokrát s o něco menším rozlišením 1920×1080 pixelů pro obě oči (960×1080 pro jedno), zorným polem asi 100° ale slibuje vyšší obnovovací frekvenci 120 Hz. Z dosud zmíněných HMD váží nejvíce, a sice 610 gramů, avšak ergonomicky jsou na druhou stranu diskutabilně nejpohodlnější. Navíc mírně nižší rozlišení v praxi však není příliš velká nevýhoda, neboť zatímco Rift a Vive používají PentTile OLED [9] [14], PlayStationVR používá RGB OLED [16]. Rozmístění



Obrázek 4: PlayStationVR (vlevo) a StarVR (vpravo)

subpixelů obou technologií jistým způsobem smazává rozdíl ve vnímání uživatelem obou variant displeje. PlaystationVR je porobně jako Rift navržen pro užívání vsedě, ale na rozdíl od produktu Oculu nelze PSVR standardně rozšířit pro použití na větší ploše či z více úhlů. Snímání brýlí je docíleno sledováním světla, které brýle vyzařují, a rovněž ovladače, které lze dokoupit, vyzařují viditelné světlo jiné barvy pro odlišení.

2.2.1.4 OSVR

Open Source Virtual Reality je projekt obsahující otevřený software a hardware [17]. Jedná se o hnutí, jehož snahou je čelit fragmentaci nově vzniklého trhu pro virtuální realitu [18]. OSVR tedy není vázáno k jediným brýlím, ačkoli ve spolupráci se společností Razer, která se zabývá tvorbou hardwaru zejména pro herní průmysl, vydali brýle OSVR HDK 1 [17] a později vylepšené HDK 2, jehož displej již je ekvivalentní Riftu či Vivu. Na trhu jsou však i specializované HMD do extrémních (venkovních) podmínek nebo určené do zábavních středisek přizpůsobené například snadno vyměnitelnými částmi.

2.2.1.5 FOVE

Další, avšak dosud nevydanou alternativou, jsou brýle Fove. Zvláštnost tohoto HMD spočívá v tom, že Fove bude sledovat pohled uživatele [19]. Tato možnost si najde řadu využití, jako například působivá interakce s prostředím (detekce očního kontaktu s postavou, či obecně využití přesného místa pohledu) nebo optimalizace vykreslování, kdy se bude vykreslovat v plném rozlišení pouze místo na displeji, kam se uživatel skutečně dívá. Fove slibuje WQHD OLED displej s působivým rozlišením 2560×1440 (tedy 1280×1440 na jedno oko) a zorným polem údajně 100° . Nemile však překvapuje pouhých 70 snímků za vteřinu, kde konkurence má 90. Váha má být 520 gramů a brýle budou vyžadovat podobné požadavky na výkon počítače, jako produkty od Oculu a HTC. Snímání pohledu má probíhat 120 krát za vteřinu a chyba měření má být pod jeden stupeň. Vzhledem k tomu, že brýle mají být kompatibilní s programy vyvíjenými pro SteamVR (tedy Vive) a OSVR [19], může se z nich (nebo alespoň z této technologie) vyklubat zajímavý úspěch.

2.2.1.6 StarVR

Na dalším zajímavém zástupci, který ještě nevyšel na trh pro koncové spotřebitele, pracuje společnost Starbreeze, která se věnuje primárně vývoji a distribuci videoher, což je pro vývoj brýlí pro virtuální realitu nepochybně výhodou. Tento projekt s názvem StarVR (Obrázek 4) je zajímavý proto, že má mít neuvěřitelné horizontální zorné pole 210° (vertikální pak 130°) [20]. Čočky mají kvůli tomu speciální tvar a displeje brýlí mají působivé kombinované rozlišení 5120×1440 pixelů (pro jedno oko tedy



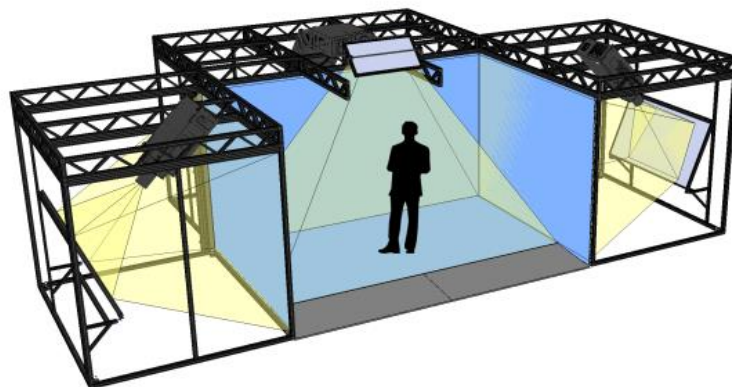
Obrázek 5: Google Cardboard (vlevo) a Samsung GearVR (vpravo)

2560 × 1440 bodů). Starbreeze udává, že displeje pokrývají 75 % lidského zorného pole, což nepochybně velmi pozitivně působí na imerzi uživatele. Nelze však nezmínit, že vykreslit působivou scénu na tak obrovské rozlišení bude vyžadovat použití extrémně výkonných grafických karet. Doufejme tedy, že v budoucnosti dojde ke kombinaci technologií různých výrobců HMD a vykreslování na displej o tomto rozlišení bude ulehčeno detekcí pohledu uživatele a renderováním nedůležitých částí displeje v mnohem nižším rozlišení. Zmíníme ještě nakonec, že váha tohoto HMD má podle dosavadních odhadů být asi 380 gramů.

2.2.1.7 Cardboard

Tento výrobek (Obrázek 5) vyvinutý společností Google je dnes ikonickým průkopníkem v dostupnosti VR. Přitom to není HMD v pravém slova smyslu – není k hlavě pevně připevněn a dokonce nemá displej. Jedná se totiž o výrobek z kartonu s dvěma čočkami a magnetem, do kterého je po jeho složení umístěn mobilní telefon s gyroskopem a pokud možno velkým displejem s vysokým rozlišením (Mobilní VR). Je to první zde zmíněný zástupce virtuální reality zprostředkované chytrými telefony. Cardboard je možné si vyrobit doma z levných materiálů a využít při tom specifikace Googlu, nebo si jej koupit za cenu do 200 korun. Jedná se tedy, pokud již člověk vlastní vhodný telefon, o velmi přístupný úvod do VR. Jedinými jeho vstupy jsou natočení telefonu pro úpravu vykreslovaného obrazu a posun zmíněným magnetem, což vytváří vstup booleovského typu.

Cardboard byl představený na vývojářské konferenci Google I/O 2014 [21] a zaznamenal od té doby obrovský úspěch. Dnes již existují desítky zařízení, která bychom mohli zařadit do jeho kategorie, a aplikací, které je využívají. Google dokonce v roce 2016 oznámil na stejnojmenné konferenci mnohem propracovanějšího nástupce – **Daydream View** [22], který je vyroben z lehkého látkového materiálu a dokonce obsahuje bezdrátový ovladač, s touchpadem, pár tlačítka a senzory pro snímání svého pohybu, aby uživateli umožnil například máchnout baseballovou pálkou či golfovou holí. Brýle si uživatel již může pevně usadit na hlavu a nemusí je držet v ruce jako Cardboard. Jejich cena se již pohybuje kolem 2 000 korun. Všechna zařízení tohoto typu pro chytré telefony mají, jak bychom asi čekali, společné to, že se do nich mobil umístí před dvě optické čočky, na displej se vykresluje stereoskopický obraz a gyroskop poskytuje telefonu informace o natočení hlavy, takže se uživatel může rozhlížet. Protože Cardboard a jeho alternativy podporují velice široké spektrum telefonů, je výsledný efekt silně závislý právě na jejich kvalitě. Mobilní telefony s nízkým grafickým výkonem či malým displejem případně jeho rozlišením poskytnou o poznání horší zážitek než zařízení vyšších cenových kategorií.



Obrázek 6: Cave Automatic Virtual Environment

2.2.1.8 GearVR

Alternativou pro Cardboard, která vznikla spoluprací společností Samsung a Oculus, je GearVR (Obrázek 5) [23]. Její cena se pohybuje kolem 2 000 korun a podporuje jen poměrně nové modely telefonů Samsung. To na druhou stranu zaručuje jistý standard kvality. Tyto telefony mají kvalitní displeje s velmi vysokým rozlišením 2560×1440 pixelů a vcelku výkonný grafický akcelerátor zajišťuje pro virtuální realitu velice důležitý vysoký, ale také stabilní, počet zobrazovaných snímků za vteřinu. Přesto mobilní VR pochopitelně nedosáhne na vykreslovací kvalitu toho desktopového, navíc absence sledování pozice hlavy je poměrně značnou nevýhodou, jak se ukázalo již při vývoji Riftu. Je ale nutné přiznat výhodu v absenci kabelu vedoucího z headsetu, což značně zpřijemňuje jeho užívání. Ceny podporovaných telefonů se aktuálně pohybují mezi 9 000 a 20 000 korun.

Brýle na své boční straně mají několik tlačítek (Zpět, Domů a nastavení hlasitosti) a touchpad, který teoreticky umožňuje stejnou preciznost ovládání, jako touchpad notebooku, ovšem fakt, že se hýbe společně s hlavou uživatele, může značně komplikovat jeho používání v aplikacích, kde je takový pohyb hlavy častý a výrazný. V takových aplikacích je vhodné touchpad využít jen pro vstup typu swipe (tedy horizontální či vertikální přejetí prstem) či pouhé poklepání. Nutno ale přiznat, že používání touchpadu brýlí poslepu značně napomáhá jeho vlis, který uživatel při používání cítí konečky prstů a tak má jakési povědomí, kde se nachází střed touchpadu. Samotné brýle váží 345 gramů (312 bez předního krytu) a i s telefonem, který se dovnitř umísťuje zapojením do svého USB-C konektoru, se tedy dostáváme na až 500 gramů.

2.2.1.9 Další

Po vzestupu Cardboard se na trhu objevilo obrovské množství alternativních řešení. Jmenujme například VR Box 3D, Homido VR Headset, BeeVR, GoClever Elysium VR, Mattel View-Master VR, Xiaomi Mi VR Play, VR Shinecon, Fibrum Pro VR a mnoho dalších. Některé slouží pouze k upevnění na hlavu, uložení telefonu a soustředění paprsků do oka, ale některé obsahují třeba i Bluetooth ovladač.

2.2.2 Alternativní zobrazení

HMD jsou elegantním řešením pro virtuální realitu, nikoli však jediným. Alternativou může být například již zmíněný systém CAVE (Cave Automatic Virtual Environment) (Obrázek 6) [5] [3] vyvinutý roku 1992 na univerzitě v Illinois. Zatímco HMD je velice kompaktní zařízení, CAVE vyžaduje, aby mu byl přizpůsoben celý pokoj. I tento systém má mnoho variant. Může jít o kompletní pokoj se čtyřmi stěnami, stropem a podlahou anebo o některou z úspornějších variant s méně plochami až po naprosté zjednodušení pouze se dvěma stěnami tvořícími roh místnosti [2]. Zmíňme také například v planetáriích běžně využívanou alternativu promítání na polokulový strop či stěny místnosti.

Zobrazení na těchto stěnách může být rovněž řešeno různými způsoby. Může jít o velké displeje či mnoho menších, anebo může jít o plátna, na která je zevnitř, avšak především zvenku, promítán stereoskopický obraz, k čemuž lze využít zrcadel. Uživatel v místnosti má nasazený brýle, které plní dvě funkce, jednak do každého oka propustí pouze vhodně polarizované světlo pro efekt trojrozměrného zobrazení a jednak je jejich poloha přesně snímána senzory, aby se promítaný obraz mohl vhodně měnit v reakci na pohyb uživatele. Vzhledem k náročným požadavkům na takovou místnost, vysoké ceně a náročnosti na sestavení se nejedná o vhodné řešení pro běžného uživatele. S takovými systémy se tedy můžeme setkat spíše v zábavních parcích či výzkumných laboratořích [24]. Přesto jim však nelze upřít, že dalece předčí náhlavní displeje ve svém dosažitelném rozlišení a zorném poli.

2.2.3 Ovladače

Jak vyplývá z textu výše, některá zařízení pro mobilní virtuální realitu podporují jako svůj vstup pouze informaci o orientaci uživatele. Takový zážitek by však mnohá ani neprošel definicí virtuální reality, kterou jsme si stanovili na začátku této kapitoly, neboť interakce uživatele s takovým světem je prakticky nulová. V této podkapitole jsou probrány další vstupní zařízení, přičemž některá umožňují interakci s virtuálním světem skutečně působivou a přirozenou.

2.2.3.1 Standardní ovladače

Jak již bylo zmíněno společnost Oculus se svými brýlemi Rift dodává ovladač konzole Xbox. Pro interakci s virtuální realitou tedy lze používat periferní zařízení, na které jsme již roky zvyklí prací či zábavou na počítači či konzoli. Ačkoli je takový ovladač nepochybně vhodnější, než použití klávesnice a myši, i tato zařízení lze použít. Netřeba však zmiňovat, že tyto nástroje nejsou vrcholem nejmodernějších výzkumů. Navzdory tomu mohou být vstupní zařízení, která známe již dlouho, pro specifické aplikace velmi vhodné. Vezměme si jako příklad simulátor řízení automobilu nebo jednoduše závodní hru ve virtuální realitě. Není těžké si představit, že ovladač v podobě volantu, pedálů a možná i řadicí páky bychom využili s výhodou.

2.2.3.2 Ovladače Vive, Touch a Phantom

Brýle Vive ve svém balení obsahují také dva ovladače (Obrázek 7), které jsou snímány stejně jako HMD v prostoru o rozloze až $4,6 \times 4,6$ metrů. Oba mají tlačítka podobná spoušti pistole, stisknutelný touchpad pro palec, tlačítka umožňující detekovat stisk ovladače v dlani a ještě dvě další tlačítka na horní straně [14]. Oculus zase vydal vlastní ovladače Touch (Obrázek 7), které lze dokoupit zvlášť v balení, které obsahuje také druhý senzor pohybu. Oba pak doporučuje umístit buďto kus od sebe před uživatele, nebo naproti sobě v závislosti na požadavcích. Jak již bylo zmíněno lze dokoupit další senzory pro zvýšení přesnosti, avšak na snímání pohybu na větší ploše Oculus stále pracuje a je zatím obsažena jen jako experimentální možnost [9]. Ovladače Oculus mají tlačítka podobná spoušti podobně jako ovladače Vive, další tlačítko pro palec a několik tlačítek na horní straně, avšak namísto touchpadu mají analogové páčky. Ty jsou pro některé vstupy sice vhodnější než touchpad, ale méně vhodné pro jiné. Jak ovladače od HTC, tak od Oculus podporují jednoduchou silovou zpětnou vazbu v podobě vibračních ovladačů. Systém Oculus je však poměrně náročný na počet portů, které vyžaduje. Pokud totiž uživatel chce používat HMD a tři senzory (což je pro technologii trackování Oculus pro použití v prostoru o velikosti pokoje vhodné), bude potřebovat čtyři sloty typu USB 3.0. Pokud chce navíc, jak bychom asi čekali, mít v počítači zapojenou ještě klávesnici a myš, potřebuje další dva USB 2.0 porty.



Obrázek 7: Ovladač HTC Vive (vlevo) a Oculus Touch (vpravo)

Vžití uživatele do virtuální reality silně umocní výstupní zařízení využívající silovou a dotykovou odezvu. Jako silová je chápána odezva, kterou uživatel vnímá pomocí svalů, kloubů a šlach. Naproti tomu dotyková odezva je ještě složitější na dosažení, neboť se jedná o reprodukci pocitu z materiálu při dotyku například hrubosti či teploty materiálu [5].

Asi nejjednodušší případ silové odezvy, se kterým se navíc setkáváme již dlouhou dobu, je vibrace ovladače. Ty podporují ovladače brýlí Vive i Oculus Touch, podobně jako ovladače herních konzolí. Daleko působivější využití silové odezvy ve virtuální realitě je však zabránění uživateli v pohybu tam, kde se nalézá virtuální překážka. To však vyžaduje poměrně složité mechanické řešení. Vhodným řešením je například použití robotické paže. Uživatel v ruce drží její konec, jehož pozici paže snímá, a pokud počítač vyhodnotí, že došlo ke kolizi s objektem, servomotory paže začnou působit proti pohybu uživatele. Historickým příkladem takového systému je GROPE-III (Obrázek 8) [5], nebo novější a kompaktnější Phantom Premium (Obrázek 8) společnosti Geomagic, která je určena pro lékařské, designérské a umělecké účely [25].

2.2.3.3 Vlastní ruce

Ovladače od HTC a Oculus jsou vskutku působivá zařízení a pro aplikace, kde uživatel neustále drží v ruce předmět, který zhruba odpovídá jejich tvaru, jsou asi nejvhodnější, neboť poskytují široké možnosti vstupu a jejich držení v takovém případě nepůsobí rušivě, právě naopak. V ostatních aplikacích by však mohlo být vhodnější použít svoje vlastní ruce jako vstupní zařízení. Tím se ostatně již řadu let, ač ne původně pro účely virtuální reality, zabývá společnost Leap Motion.

Leap Motion Orion je nový softwarový a hardwarový produkt, jehož účelem je okamžité a přesné čtení gest rukou speciálně pro virtuální realitu [26]. Tento systém byl ve své původní verzi určen pro uložení na desce stolu a uživatel gestikuloval nad jeho senzorem. Po představení vývojářské verze Riftu byl však umisťován přímo na HMD. To pro čtení gest představuje nové problémy, neboť ruce uživatele nejsou snímány z tak výhodného úhlu. Hlavní problém při užívání ale tkvěl v tom, že gesta byla snímána jen v omezeném prostoru před senzorem. Právě tyto problémy má Orion vyvíjený speciálně pro VR úspěšně adresovat. Orion má začít číst gesta rychleji, má být přesnější a má být schopný daleko lépe oddělit ruku uživatele od pozadí. To vše je v režii softwarové části produktu. Hardwarový snímač má být vestavován do HMD.

Handpose je projektem z dílny Microsoftu [27]. Používá jedinou hloubkovou kameru zařízení Kinect konzole Xbox One a podobně jako produkt od Leap Motion je flexibilní z pohledu umístění kamery a zotavení z nepovedeného sledování. Vzhledem k tomu, čeho je Kinect schopný v rámci detekce celé postavy ale zůstává nezodpovězenou otázkou, zda Microsoft plánuje vyvinout řešení právě pro snímání celého těla.



Obrázek 8: GROPE III (vlevo) a Phantom Premium (vpravo)

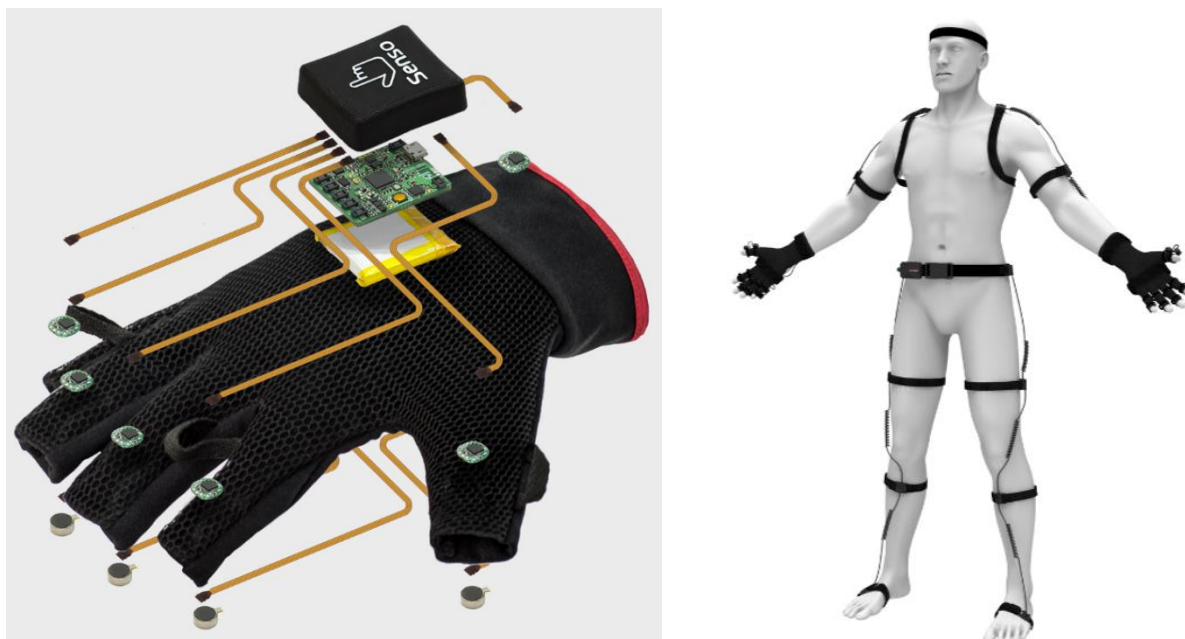
RinkVR od společnosti Samsung pro jeho GearVR je ovladač nasazený na ruce uživatele, který dokáže snímat svou pozici, rotaci a pohyb prstů. V praxi je detekce gest velmi omezená [28], ale ovladač je stále ve vývoji, takže můžeme očekávat zlepšení. Navíc se jedná o zařízení pro mobilní VR, což je kategorie, která představuje nízkou cenovou náročnost. Vzhledem k tomu, že můžeme očekávat poměrně nízkou cenu zařízení a zlepšení detekce gest, jedná se o velmi zajímavý výrobek.

Manus VR je řešením pro snímání gest prostřednictvím rukavice, které je využíváno NASA pro trénink astronautů. Ačkoli v minulosti měla mírné problémy s vyšší latencí a chybělo jí sledování pozice zápěstí, což mělo být řešeno stejným optickým systémem, který používá Vive [29], dnes se specifikace zatím stále nevydaného produktu chlubí odezvou pod 5 milisekund avšak senzory polohy IMU (Inertial Measurement Unit) [30]. Ty obecně bývají vnímány jako nepříliš vhodné řešení, neboť svou pozici určují pouze na základě akcelerometru, gyroskopu a magnetometru a nikoliv na základě globální pozice, jako v případě optických senzorů. Nepřesnosti ve sledování se zejména mohou nahromadit časem při používání. Tuto technologii ale využívají i níže popsané ovládací prvky a jejich tvůrci nás většinou ubezpečují, že implementovali algoritmy a techniky, jak tyto nevýhody prakticky odstranit [31] [32]. Rukavice mají dále mít na hřbetu ruky vibrační motor pro zpětnou vazbu a baterie mají vydržet 3 – 6 hodin provozu.

GloveOne je další dosud nevydanou rukavicí pro sledování rukou, jejíž hlavní síla spočívá v umístění vibračních jednotek do konečků svých prstů, čímž simuluje pocit dotyku a interakce s prostředím. Rukavice ve své první verzi nedisponovala sledovacím zařízením, ale její tvůrci vyvinuli senzory využívající IMU senzory, jejichž nepřesnosti opravují s využitím centrální jednotky umístěné na hrudi uživatele [33]. To má za následek přesné sledování pozice rukou i torza uživatele. Poměrně netradiční však je, že tato verze rukavice nedetekuje gesta, ale pouze využívá pohybu prstů podobnému štípání pro interakci s virtuálními objekty

PowerClaw je další z rukavic, která se tentokrát soustředí pouze na výstup pro uživatele a detekci pozice rukou a jejich gest ponechává plně na řešeních třetích stran. Zařízení bylo představeno na veletrhu Gamescom 2016 a zajímavá je tím, že na konečcích prstů má umístěny malé motory a termoelektrické buňky, které umožňují precizní vibrace, elektrošoky a dokonce působivou změnu teploty [34]. Rukavice tak dokáže simulovat pocit dotyku ledové kostky, plamene, či bodání jehlou. Kromě absence sledování polohy a gest je však další nevýhodou kabel obstarávající napájení. Zklamáním je ale především to, že tento projekt možná pomalu zaniká.

Senso (Obrázek 9) je však vyvíjená alternativa, která rovněž slibuje simulaci tepla a chladu. Dále má zprostředkovat dotykovou zpětnou vazbu prostřednictvím vibrací pěti motorů na každé ruce a poloha rukou má být sledována bez použití kamer sedmi již známými senzory IMU na každou ruku,



Obrázek 9: Senso Glove (vlevo) a Perception Neuron (vpravo)

kteří mají zajistit šest stupňů volnosti. S využitím vlastního snímacího softwaru mají předejít znatelným nepřesnostem způsobeným použitím IMU jednotek [32]. Zařízení údajně zpracovává 150 měření za vteřinu, její latence má být pod 10 milisekund prostřednictvím Bluetooth Low Energy a baterie má vydržet 10 hodin používání.

2.2.3.4 Celé tělo

Umožnit rukám uživatele, aby přímo interagovaly s virtuálním prostředím je velmi velký krok směrem k přirozené interakci člověka se strojem. Některé aplikace však mohou s výhodou těžit ze snímání celého těla uživatele. Již v minulosti jsme byli svědky částečného úspěchu senzoru Kinect a můžeme doufat, že společnost Microsoft využije jeho potenciál pro interakci s virtuální realitou a rozšíří tak svůj projekt Handpose. V současnosti se však rýsují alternativní možnosti pro snímání celého těla uživatele. Největší problémy, se kterými se níže uvedená řešení musí potýkat, jsou přesnost detekce a latence. Michael Abrash uvádí, že při hraní videohry a sledování jejího obrazu na monitoru se setkáváme běžně s latencí 50 milisekund, avšak ve virtuální realitě je z jeho osobní zkušenosti nad 20 milisekund příliš dlouhá doba. Uvádí dále, že výzkumy naznačují, že spodní hranice je 15 či dokonce 7 ms [35]. To platí zejména pro náhlavní displeje, pro vstupy získané snímáním celého těla může být tato hranice mírně výše. Avšak není třeba zmiňovat, jak rušivý a nepříjemný efekt ve virtuální realitě způsobí, když svoje virtuální tělo uživatel vidí na jiném místě či znatelně zpožděně oproti tomu skutečnému.

MVN Awinda je jedno z hardwarových řešení společnosti Xsens určené primárně pro zachytávání pohybu animací postav pro herní a filmový průmysl, avšak již bylo použito pro snímání těla uživatele ve VR. Systém se skládá ze 17 bezdrátových IMU sensorů a jejich baterie mají vydržet 6 hodin provozu [36]. Problémy však mohou být latence tohoto zařízení 30 milisekund, cena, která přesahuje 300 000 korun, a jako u všech systémů s IMU sensory, nepřesnosti ve snímání při pohybech jako je běh.

Perception Neuron (Obrázek 9), jako další ze zařízení vyvinutých primárně pro snímání pohybu pro účely animace ve hrách a filmu, je softwarové a hardwarové řešení, které se skládá z až 32 sensorů. Každý má trojosý akcelerometr, gyroskop a magnetometr, a tak poskytuje devět stupňů volnosti. Používá tedy sice sensory IMU, ale tvůrci se opět pyšní tím, že při správné kalibraci jsou nepřesnosti minimální. Varují však, že se mohou časem nastrádat. Sensory je možné libovolně umístit na tělo herce

a tak lze snímat pohyb například pouze ruky pomocí pouhých tří sensorů, anebo celého těla včetně prstů s využitím všech 32 jednotek. Cena tohoto systému je přijatelnější než v předchozím případě, ale stále na 40 000 korunách za systém o 32 neuronech není dostatečně nízko pro většinu spotřebitelů pro domácí použití. Vzhledem k tomu, že součástí senzoru je magnetometr, musíme se mít na pozoru před jejich rušením magnetickým polem. Při využití tohoto systému pro účely virtuální reality se však musíme opět ptát na zpoždění systému. Výpočty uvnitř senzoru mají trvat až 13 ms, výpočty v počítači až 5 ms a k tomu ještě musíme připočíst čas nutný k přenosu dat ze sensorů do PC (který závisí na tom, zda je použit pomalejší, ale bezdrátový, přenos přes Wi-Fi, nebo rychlejší možnosti přes USB či uložení lokálně na SD kartu) a k vykreslení scény [37].

PrioVR je další zařízení které využívá IMU sensorů pro snímání celého těla uživatele. Součástí zařízení jsou dva ovladače, jeden do každé ruky se spouští, několika tlačítka a analogovou páčkou. Toto zařízení má být v budoucnu určené pro domácí použití, ale navzdory tomu cena jeho vývojářské verze zatím přesahuje 30 000 korun [38]. V demonstračních videích tohoto výrobku jsou bohužel také znatelné nepřesnosti sensorů a můžeme se nechat jedinečně překvapit, zda budou do vydání zákaznické verze odstraněny.

VicoVR je velmi ambiciózní projekt, který slibuje snímání celého těla reprezentovaného kostrou s 19 klouby pro mobilní virtuální realitu. Nemá při tom k tomu využívat žádné senzory připevněné k uživateli, nýbrž hloubkový sensor umístěný před uživatelem. Slíbují snímání až dvou lidí současně a to bez dalšího výpočetního zařízení jako je počítač. Má být kompatibilní se zařízeními Android a iOS a dostupný v roce 2017. Problémem ale může být opět latence. Sledování kostry údajně trvá asi 15 ms a bezdrátový přenos asi 6,5 ms [38]. Pokud k těmto 21,5 ms přičteme ještě dobu nutnou pro výpočet kreslení, už můžeme zacházet za vhodnou hranici.

Tesla Suit (Obrázek 10) je stále vyvíjený oblek na celé tělo, který využívá elektrických impulsů přímo do kůže pro simulaci dotyku, změny teploty a síly. Tato technologie pochází původně ze zdravotnictví, kde byla využívána pro rehabilitaci a úlevu od bolesti. Oblek má údajně být schopný i simulace změny teploty, vodě odolný, vypratelný a s počítačem se bude spojoval bezdrátově [39]. Tesla Suit má být schopen simulovat vánek, objetí, explozi či výstřel ze zbraně. Výpočetní jednotka umístěná na starší verzi obleku obsahovala mimo jiné čtyř-jádrový procesor o frekvenci 1 GHz, 1 GB paměti RAM a baterii o kapacitě (elektrickém náboji) 10 000 mAh a běží na ní proprietární software Tesla OS. Oblek má být modulární a jeho prototyp disponuje 52 výstupními kanály [40].

2.2.3.5 Další zařízení

Simulátory určené pro výcvik či zábavu mohou pro ještě větší efekt využívat HMD. Zařízení jako simulační křesla s volantem a pedály **Atomic A3** [41] či simulátory společnosti **Vesaro**, pak uživatele přesunou do extrémně uvěřitelného virtuálního prostředí. Hydraulické písty dokážou velmi rychle pohybovat s celou konstrukcí, takže pocit vozu skákajícího po nerovné cestě je téměř dokonalý. Jedná se však o náročné systémy s cenou i dalece přesahující 150 000 korun.

Existují však produkty ve vývoji, které jsou určeny pro domácí použití. Například samo-otáčecí křeslo **Roto VR** je velmi zajímavým řešením, které má brzy vyjít na trh a jehož cena se dle zvolené verze pohybuje asi mezi 18 000 až 32 000 korun [42]. Další zařízení, které zve uživatele k usednutí, se jmenuje **VR GO** [43]. Jde o sedadlo, jehož spodní část má vejcovitý tvar, a tak umožňuje uživateli změnu těžiště naklonění. Tuto změnu stavu okamžitě transformuje pro účely pohybu ve virtuálním prostředí. Váží 4 kilogramy, baterie vydrží asi 5 hodin používání a pro odolnost je vyrobeno z kompozitu, užívaného při výrobě letadel. Velmi podobný název – **VRGO** nese přenosný počítač s baterií firmy Zotac zhotovený jako batoh, určený pro virtuální realitu [44].

Další kategorií zařízení pro užití s virtuální realitou jsou systémy, které uživateli umožní neomezený pohyb. Produkty jako **Virtuix Omni** (Obrázek 10) [45], **Cyberith Virtualizer** [46], **Kat**



Obrázek 10: Tesla Suit (vlevo), Virtuix Omni (uprostřed) a InfinaDeck (vpravo)

Walk [47] nebo **WizDish** [48] toho dosahují pomocí povrchu s velmi nízkým koeficientem tření, po kterém uživatel upevněný v pase v konstrukci chodí na místě. Pro detekci chůze při tom používají systémy rozmanité způsoby. Například Omni používá IMU senzory umístěné v botách se speciální podrážkou a Wizzish zase mikrofony pod podlahou, které snímají hluk vydávaný pohybem nohou. Omni a Virtualizer ve své konstrukci obsahují obruč s postrojem ve výšce pasu, v němž uživatel stojí a pevně jej drží a umožňuje mu tak chůzi i běh. Díky tomu, že tuto obruč má Virtualizer umístěnou v polohovatelných sloupech, dovoluje také skrčení a výskok. To umožňuje i Kat Walk, který navíc upevnění uživatele neřeší pomocí obruče, ale postroje upevněného na zádech uživatele, takže umožňuje volnější pohyb. Tyto výrobky zatím nejsou dostupné na trhu, ale míří na domácí použití. Kat Walk má dokonce umožňovat do sebe namísto postroje zavěsit křeslo, takže může fungovat i jako nábytek.

Poněkud jiný přístup k umožnění pohybu, než výrobky výše popsané zvolili výrobci zařízení **InfinaDeck** (Obrázek 10). Jedná se o všesměrový běžecký pás, který se pohybuje podle tlaku vyvinutého náklonem uživatele do upevňovacího postroje. Toto řešení sice není stále pocitově totožné se skutečnou chůzí, ale v některých případech, jako je zejména změna směru, poskytuje daleko uvěřitelnější pocit, než zařízení v minulém odstavci využívající nízké tření. Pohyb pásu prototypu sice dosahuje rychlosti nejvýše asi 6,5 kilometrů za hodinu (1,8 m/s) a chůze na něm je obtížnější s nasazeným náhlavním displejem, ale technologie je ještě ve vývoji. Tvůrci InfinaDeck si navíc uvědomují, že jejich výrobek bude, alespoň v jeho současné podobě, poměrně drahý a velký a necílí tak prozatím na domácí použití, ale spíše na využití průmyslové, vojenské a v zábavních centrech [49].

3 Modely

Jednotlivé prvky používané při vývoji mnoha projektů bývají označovány poněkud obtížně přeložitelným termínem, a sice jako takzvané **assety**. Jejich asi nejtypičtějším příkladem pro interaktivní 3D aplikace jsou polygonální modely (případně rozšířené o vlastnosti jejich povrchu, animace a chování). Je však nutné zdůraznit, že to je jen příklad objektů, které tímto pojmem můžeme označit. Tento termín má totiž obecně daleko širší záběr. Jako assety můžeme zjednodušeně označovat téměř libovolné části projektu, jako může být dále například modul pro umělou inteligenci nebo uživatelské rozhraní. Nadcházející kapitola pojednává o možnostech reprezentace 3D modelů a možnostech jejich tvorby.

3.1 Reprezentace

Polygonální modely jsou sice pro typ aplikace, která je předmětem této práce, sice nepopíratelně nejvhodnější volba – vykreslovací řetězec je navrhnutý speciálně pro jejich zobrazení – nicméně je vhodné zmínit i další alternativní možnosti reprezentace trojrozměrných objektů. Kromě povrchové reprezentace jsou nejrozšířenějšími a nejužitečnějšími dalšími možnostmi voxelová a CSG reprezentace objektů [50].

Hraniční reprezentace objektu polygonální sítí je neoddiskutovatelně nejrozšířenější způsob popisu trojrozměrných objektů. Jednotlivé body modelu (*vertices*, avšak jeden *vertex*) jsou spojené hranami (*edges*) a ty tvoří stěny (*faces*, *polygons*). Ačkoli je základní plochou trojúhelník, často se při tvorbě modelů v této reprezentaci využívají čtyřúhelníky. Ty totiž modelu dodají správný „proud hran“ (*edge flow*), což má pozitivní dopad jak na vzhled modelu, tak na náročnost jeho tvorby a výsledné geometrie. Tato reprezentace je dnes považována za jakýsi standard, který je hardwarově i softwarově nejvíce podporován a nejširěji využíván.

Voxelová reprezentace rozděluje prostor mřížkou, ve které má každá buňka definovanou hodnotu. Pokud tato hodnota představuje pouze barvu, je voxelový objekt vlastně takový trojrozměrný obrázek. To odráží už samotný název jednoho prvku mřížky – *voxel*. Jedná se totiž o složeninu slov *volumetric* a *element*, tedy *objemový prvek*, podobně jako slovo *pixel* vzniklo složením slov *picture element*, tedy *prvek obrázku*. Výhoda této reprezentace spočívá v tom, že má objekt definovaný celý svůj objem, proto je používán pro uložení a zobrazení vědeckých či lékařských dat, kde má ostatně tato reprezentace také svůj původ. Mimo to se ale užívá pro fyzikální simulace, například šíření tepla tělesem, nebo pro práci s terénem, neboť jej následně lze snadno deformovat. Slabšími stránkami je ale paměťová náročnost a problematická animace voxelových objektů.

Konstruktivní geometrie těles (Constructive Solid Geometry – CSG) využívá pro vyjádření objektů základní geometrické tvary (jako krychle, koule, kužel, kvádr,...), mezi kterými stanovuje vztahy na základě booleovských množinových operací (tedy sloučení, průnik a rozdíl) a tím je staví do *n*-árního stromu. Tento způsob reprezentace se kvůli přesnému matematickému vyjádření jednotlivých základních tvarů může hodit pro automatizovanou výrobu a kvůli jednoduchosti vyjádření modelu může být vhodný i pro procedurální generování některých objektů. Je však zřejmé, že je tato reprezentace vhodná jen pro omezenou podmnožinu modelů a aplikací.

Implicitní plochy neboli **metaballs** (také Implicit Solid Modelling nebo Blobs) [51] je už méně známá metoda, jejíž základem jsou body v prostoru představující středy koulí. Tato metoda za svou existenci vděčí J. F. Blinnovi, který ji původně vyvinul pro účely simulace rozložení elektronové hustoty mezi atomy vodíku. Tato reprezentace umožňuje popsat poměrně kompaktním způsobem

i objekty, jejichž popis trojúhelníkovou sítí by byl velmi náročný z pohledu tvorby i potřebného paměťového prostoru.

Dalšími možnostmi jak reprezentovat trojrozměrné modely je také využít **NURBS křivky** (NonUniform Rational B-Splines), které umožní jednoduše popsat zaoblený povrch, jinak popsany velkým množstvím trojúhelníků. Poněkud exotičtější je využití **Surfelů**, což jsou orientované rovinné plošky popisující povrch objektu. Na závěr ještě zmíníme rozšířenější způsob reprezentace objektů pomocí **mračen bodů**, získaných například pomocí 3D skenerů.

3.2 Tvorba

Podle charakteru modelu můžeme volit různé způsoby jeho získání. Pokud vyžadujeme velmi detailní model skutečného objektu, může být ideální způsob jeho získání **3D sken** [52]. Dnešní 3D skenery dokážou v poměrně krátkém čase vytvořit velmi podrobné modely. Existují stacionární i ruční skenery, které dokáží během pár minut totéž, čím by grafik strávil hodiny. Je ale otázkou, v jaké aplikaci požadujeme tak precizní modely. Většinou se spokojíme s tvarem jednodušším, na kterém si ani nevšimneme rozdílu v kvalitě a ušetřený výkon ve vykreslování uvítáme.

Pokud vyžadujeme velké množství modelů, může být správnou odpovědí jejich **procedurální generování**. [53] Modely mohou být kompletně výsledkem algoritmu, nebo můžeme vytvořit tvary, které program modifikuje a využije při tvorbě nových modelů. Často jsou takto tvořeny stromy, rostliny, keře nebo dokonce gotické stavby. Běžné je též generování terénu.

Výše zmíněné přístupy mají samozřejmě své přednosti, ale mnohé modely vytváří umělci **ručně** v programech tomu určených. Autor tak má naprostou kontrolu nejen nad estetickou stránkou modelu, ale i nad tou praktickou, jako je složitost modelu, myšleno počet trojúhelníků, jež jej tvoří. Na tomto místě je vhodné zmínit ty nejvýznamnější nástroje, které má umělec pro tvorbu k dispozici.

3.2.1 Modelovací nástroje

3D Studio Max je jedním zástupcem těchto programů. I přes to, že jeho grafické uživatelské rozhraní (GUI) je přívětivé i pro nováčky, obsahuje mocné nástroje pro tvorbu modelů a i jeho možnosti pro tvorbu animací a nanášení textur jsou dostatečně silné pro profesionální užití. To je ale znát na ceně tohoto softwaru, která v závislosti na konkrétní volbě produktu může přesáhnout osmdesát tisíc korun [54], a je tedy skutečně dost vysoká na to, aby si vývojářská studia s běžným rozpočtem hledala alternativní řešení. Program je nicméně vhodný nástroj pro tvorbu libovolných assetů pro 3D aplikaci.

Maya je další velmi rozšířený nástroj pro tvorbu trojrozměrných modelů, jehož síla se může měřit s výše zmíněným programem. Jedná se také o profesionálně používaný program jak v oblasti vývoje her a interaktivních aplikací tak ve filmovém průmyslu. Ačkoli je cena tohoto softwaru prakticky stejná jako u minulého produktu [55], jedná se ostatně o velmi podobný nástroj a dokonce od stejné firmy, existuje jeho odlehčená verze nazvaná **Maya LT**, která však obsahuje všechny nástroje potřebné pro tvorbu kvalitních assetů pro herní průmysl. Cena této okleštěné verze tomu odpovídá a pro odpovídající časové období se pohybuje kolem třinácti tisíc. Výhoda plné verze Mayi oproti 3DS Maxu spočívá mimo existence odlehčené verze například také v možnosti skriptovat náročnější operace v jazyce Python, nebo vlastním jazyce MEL. Maya je též známá pro své velmi mocné animační nástroje. Výhody Maxe jsou zase přívětivé uživatelské rozhraní a dobrý proud práce (workflow) [56].

Blender je program velmi rozšířený nejen mezi lidmi, pro něž je modelování koníčkem, ale rovněž v (zejména nezávislých) vývojářských studiích. Jedná se o projekt s otevřeným zdrojovým kódem (Open Source), který je tedy dostupný zcela zdarma a to i pro komerční užití. Přitom je díky své obrovské komunitě stejně mocný, jako oba výše zmíněné komerční produkty [57] a to dokonce včetně podpory skriptování v jazyce Python. Jeho rozsáhlá komunita také poskytuje mnoho příležitostí, kde

spolehlivě najít řešení případného problému, který se může naskytnout při práci, což ve výsledku odpovídá technické podpoře již zmíněných programů. Pro úplného začátečníka může být v modelovacím prostředí mírně matoucí jeho uživatelské rozhraní, ale je možné se s ním s trochou snahy poměrně brzy seznámit a odhalit tak silné stránky v jeho původní složitosti. Vzhledem ke všem kladům tohoto projektu a k tomu, že zmíněná přístupová bariéra již byla v mém případě překonána, zvolil jsem Blender jako nejvhodnější nástroj pro účely této práce. Zmiňme však ještě velmi stručně dva důležité modelovací nástroje.

SketchUp je program vyvíjený společností Google, ovšem ačkoli dokáže exportovat modely ve formátu typickém pro program 3DS Max, je navržený zejména pro modelování budov. Práce v něm může být sice rychlá, ale jeho nástroje silně pokulhávají, pokud bychom chtěli vytvářet modely některých složitějších objektů.

AutoCAD je oproti tomu navržený přesně pro modelování přesných složitých tvarů technických součástí, ale ačkoli je velmi dobrý v tom, pro co byl vytvořen, ani on není nejšikovnější nástroj pro naši konkrétní požadovanou aplikaci.

4 Návrh aplikace

Cíl vytvořené aplikace, kterým bylo vhodně demonstrovat vytvořené assety, bylo pochopitelně nutné brát v potaz při jejím návrhu. Proto je jádrem návrhu schopnost uživatele snadno a rychle přistoupit ke stěžejním objektům. Toho mělo být dosaženo intuitivním využitím ovladačů, které drží v ruce. Bylo tedy dále nutné při návrhu brát ohled na technické řešení těchto ovladačů a jejich možné vstupy.

4.1 Uživatelské rozhraní

Součástí návrhu aplikace jako celku je způsob jakým uživatel interaguje s virtuálním světem. Vhodné uživatelské rozhraní může rozšířit jeho možnosti a zvýšit pohodlí aniž by působilo rušivě.

4.1.1 Tradiční prvky pro VR

Pro aplikace ve virtuální realitě je dnes samozřejmostí, že implementují nějaký způsob, jak uživateli signalizovat některé informace, jakými jsou například pozice a orientace předmětů snímaných systémem nebo prostorové limity skutečného prostoru, který má uživatel k dispozici. Již dnes existují poměrně zaběhnuté způsoby, jak tyto informace uživateli zobrazit a pro účely navrhované aplikace nebyl důvod tyto konvence měnit.

Ve virtuálním prostředí jsou tedy ovladače, které uživatel drží v ruce, vykresleny jako 3D modely na takovém místě a v takové orientaci, jak jsou snímány senzory, a dále jsou hranice místnosti renderovány jako obdélníkový obrys na podlaze, přičemž při přiblížení k hranicím je dále vykreslena mříž reprezentující konec prostoru, ve kterém je bezpečné se pohybovat. Poslední důležitý prvek dnes již tradičně využívaný v aplikacích pro VR je setmění displejů, pokud se hlava uživatele ocitne ve virtuálním objektu. Neboť však naprostá slepota nepůsobí uživatelsky právě přívětivě, lepším návrhem je pouze ztemnění obrazu anebo jeho silná vinětace (ztemnění okrajů obrazu s postupným zesvětlováním k jeho středu).

Důležité je také rozhodnutí zda (případně jakým způsobem) umožnit uživateli pohyb ve scéně. Schopnost využít pouze prostor reálného pokoje je značně limitující jak ze strany vývojáře tak uživatele, a proto byla vyvinuta řada způsobů, jak se ve virtuálním prostředí pohybovat. Vzhledem ke komplexnosti tohoto problému a tomu, že se této problematice kvalitně a podrobně věnují jiné práce, se zde však nebudeme věnovat jejich výčtu a pouze popíšeme zvolenou metodu, kterou je teleportace na krátké vzdálenosti. Teleportace je ve VR velmi často využívaným konceptem, který jednoduše obchází limitace velikosti běžného pokoje. Samotný proces teleportace je často řešen tak, že jakmile uživatel určí cílové místo, displej jeho HMD se rychle setmí, změní se jeho pozice a následně dojde k rychlému obnovení zraku. Toto řešení má výhodu v tom, že nezpůsobuje nevolnosti, avšak může být mírně dezorientující jednoduše tím, že uživatel na okamžik ztratí zrak a následně se objeví na jiném místě. Z tohoto důvodu byla zvolena mírně odlišná implementace, kdy zrak je po celou dobu přesunu zachován a přesun samotný je velmi rychlým pohybem na cílové místo s velmi vysokým zrychlením při startu a zpomalení. Ačkoli na papíře tento způsob působí poměrně nevěrohodně, co se týče způsobování nevolnosti, v praxi se ukázalo, že s ním většina uživatelů nemá problém a ti co pocíťovali mírný nekomfort si velmi rychle zvykli. Protože tato varianta pohybu nestmívá obraz, nezpůsobuje dezorientaci.

Ovládání aplikace bylo navrženo s ohledem na design ovladače Vive. Ten disponuje třemi dominantními vstupy (Obrázek 11) a sice spouští (1 na obrázku), kruhovým touchpadem (2) a tlačítkem detekujícím stisk ovladače v ruce (3), které bude dále v textu označováno jako *grip*.



Obrázek 11: Hlavní vstupy ovladače Vive (vlevo) a vytvořené kruhové menu (vpravo)

4.1.2 Přístup k hlavním objektům

Způsobů, jak uživateli zpřístupnit objekty k interakci je celá řada. Velmi intuitivní je například jejich rozmístění po scéně, ve které se pohybuje. Tyto objekty je možné sebrat a nosit s sebou nebo si je umístit do inventáře, jehož konkrétní řešení může mít opět mnoho podob. Inventář je možné vyřešit jako vlastní menu, kam si člověk objekty odloží a odkud je později vytáhne. Alternativně si uživatel může objekty umístit na tělo, jako třeba na opasek. Tento způsob by však narazil na komplikace v momentě, kdy se pas uživatele ve skutečné realitě nachází na jiném místě než ten v realitě virtuální. Tento stav může být poměrně častý, neboť systém snímá pouze pozici ovladačů a hlavy, a tak je odhad umístění pasu poměrně omezený, a to zejména v případech, kdy se uživatel naklání a při jiných pohybech, kdy hlava a pas (či trup při umístění objektu na hrudník) nejsou zarovnány ve svislé přímce. Systém sbírání objektů ze scény a umísťování do inventáře obecně je však vcelku omezující z toho pohledu, že je objekty nejprve nutné nalézt a sebrat, což je pro účely vytvořené aplikace nevhodné. To lze vyřešit umístěním všech objektů do inventáře, ale přetrvává problém hledání pasu uživatele. I při řešení inventáře ne pomocí umísťování objektů na tělo, ale pomocí vlastního menu, vyvstává nutnost do tohoto menu přistupovat, tvořící jistou zbytečnou bariéru v uživatelském rozhraní a to jednak kvůli otevírání vlastního menu a dále kvůli nutnosti věnovat výběru objektu z inventáře plnou pozornost.

Tyto a další důvody vedly k návrhu manažera objektů přístupného pomocí jednoho z hlavních vstupních prvků Vive ovladače, kterým je kruhový touchpad pod palcem. Navržený způsob přístupu k objektům zobrazí při dotyku touchpadu kruhové menu kolem něj (Obrázek 11). Pohybem palce po jeho ploše jsou vybírány ty položky, které jsou nejbližší k pozici prstu. V případě velkého množství položek byl navržen systém víceúrovňového menu, avšak počet úrovní, do kterých je možné se zanořit, by měl být co nejmenší. V ideálním případě by zanoření nebylo nutné, avšak lze předpokládat, že by v takovém případě byl počet segmentů příliš vysoký ať už z pohledu přehlednosti či obtížnosti výběru požadované položky. S dvouúrovňovým menu (jak testy potvrzují) uživatelé nemají problém, avšak můžeme očekávat, že tři úrovně zanoření by již nesly neúnosnou zátěž v navigaci. Stav, ve kterém by bylo nutné mít menu takto komplexní, skoro jistě značí špatný návrh či nevhodnou volbu prostředků. Jinými slovy by v takovém případě nebylo vhodné použít výše popsané kruhové menu. Jednotlivé položky menu ze zřejmých důvodů není vhodné mít řešeny textově. Obrázek znázorňující objekt, k němuž má tlačítko vazbu, je daleko vhodnější.



Obrázek 12: Ilustrace zobrazení plného a poloprázdného zásobníku

4.2 Modely

Nejdůležitějším aspektem navržené aplikace jsou předměty přístupné přímo z kruhového menu. Aby však tyto objekty dávaly smysl a získaly kontext, je nutné vytvořit scénu, ve které se bude uživatel pohybovat a tyto předměty využívat. Návrh předmětů, které uživatel ovládá, plyne z poznatků, jež autor práce nabyl z vlastních zážitků a pozorování ostatních ve virtuální realitě. Ukázalo se, že i lidé bez záliby ve sportovní střelbě či akčních hrách nebo filmech, se baví střelbou ve virtuálním prostředí a že interakce v něm formou úderů do objektů a jejich házení je rovněž univerzálně zábavná.

4.2.1 Scéna

Scéna splňující požadavky v úvodu této podkapitoly by tedy měla obsahovat objekty, které budou s předměty z hlavního menu vhodně interagovat. Vzhledem k charakteru předmětů, které jsou jádrem aplikace, k tomuto účelu postačí krabice fyzikálně uvěřitelně reagující na akce uživatele.

Scénu samotnou je vhodné ideově koncipovat jako hřiště. Měla by být dostatečně velká, aby se v ní uživatel necítil stísněně a scéna mu pro jeho akce poskytla dostatek prostoru, ale současně by neměla být zbytečně rozsáhlá, aby přesun po ní nenudil a nepůsobila prázdně. Vhodným prvkem pro scénu je vertikálnost, která jí umožní pohodlně vměstnat na menší prostor více zajímavostí a současně tak pozitivně působí na její členitost a vzhled. I přes vertikálnost by však scéna měla obsahovat převážně rovné plochy pro chůzi, neboť to je typ podlahy, po které se bude uživatel pohybovat ve skutečné realitě.

4.2.2 Střelné zbraně

Podstatným aspektem střelné zbraně ve VR je, jak by se dalo čekat, uspokojivá interakce s prostředím při výstřelu. Při zásahu objektu ovlivněného fyzikou by na něj měla zapůsobit síla kulky a i při zásahu statických objektů, jako je třeba podlaha, by se měl vykreslit například částicový efekt, aby měl uživatel stále vizuální odezvu na své akce.

Kromě toho je důležité umožnit intuitivní ovládání. Vzhledem k tomu, že jedním ze vstupů ovladačů Vive je tlačítko pod ukazováčkem velmi podobné spoušti pistole, byl tento vstup jasnou volbou pro vstup se sémantikou výstřelu. V případě, že ve zbrani dojde střelivo, je umožněno vložení nového zásobníku stiskem posledního dosud nevyužitého z hlavních tlačítek a sice *gripu*.

Klíčové dále je, uživateli poskytnout přehled o kapacitě zásobníku zbraně, což často předejde zmatení, když zbrani dojde střelivo. Informaci o stavu zásobníku, je možné znázornit několika způsoby, například zobrazením čísla reprezentujícího počet nábojů vedle zbraně. Pokud je však možné vyhnout se zbytečnému grafickému uživatelskému rozhraní, je to v zájmu imerze rozhodně dobrá praktika. Proto je vhodné takový číselník zakomponovat do grafického designu zbraně. Alternativním způsobem, který byl zvolen v navržené aplikaci, je umožnit uživateli přímo vidět kolik nábojů mu zbývá, díky

zásobníku s průhledem na patrony (Obrázek 12). Tímto způsobem, může mít uživatel při střelbě neustále na očích přirozenou a intuitivní reprezentaci stavu zásobníku.

Ačkoli v základní verzi navržené aplikace měla být ze střelných zbraní pouze pistole, byl využit prostor pro možná rozšíření a byly implementovány další dvě střelné zbraně. První z nich je upilovaný brokový troják. Jak název napovídá, jedná se o brokovnici se třemi zkrácenými hlavními. Brokovnice byla zvolena proto, aby uživatel mohl jediným výstřelem působit na objekty ve scéně větší silou, neboť každý brok, který opustí po výstřelu hlaveň má sice nižší energii než jedna kulka z pistole, ale jejich energie vynásobená jejich počtem jediného výstřelu daleko převyší schopnost pistole. Zkrácená verze zbraně byla zvolena proto, aby byla brokovnice jednoruční zbraní a důvod pro tři hlavní byl ten, že takto nízký počet možných výstřelů spolu s nutností záměrného stisku tlačítka pro každý z nich uživatel bez problému udrží na paměti, a tak fakt, že nemá kapacitu zásobníku neustále zobrazenou, nepředstavuje problém. Skutečnost, že broky po výstřelu mají jistý rozptyl, což zjednoduší zásah na větší vzdálenost i přes méně přesná mířidla, je pak samozřejmostí.

Třetí navrženou střelnou zbraní je jednoruční revolverový raketomet. Ten umožní působit současně na ještě větší množství objektů ve scéně ještě větší silou. Stav zásobníku je uživateli komunikován díky tomu, že jsou rakety ve zbraní rovněž viditelné.

4.2.3 Meč a vrhací předměty

Jak bylo naznačeno v úvodu této podkapitoly, v raných fázích vývoje bylo zjištěno, že se testéři baví házením předmětů a údery do nich. Toho bylo využito při návrhu ostatních předmětů, kterými jsou japonský meč a vrhací hvězdice. Dále byl navržen palcát pro možnost volby mezi přesnějším mečem a poněkud hmotnější zbraní a granát do kategorie vrhacích předmětů, který kombinuje ideu vrhacích hvězdic a raketometu. Byla vytvořena i jeho silnější verze s mocnější explozí avšak vyšší hmotností a je tedy obtížnější jej hodit na větší vzdálenost.

Úderné zbraně není třeba nijak ovládat, jediná jejich interakce se světem spočívá v kolizi s ním. Pro vrhací zbraně však bylo nutné navrhnout způsob hodu. Je zřejmé, že by směr a síla hodu měla vycházet z pohybu paže uživatele, respektive ovladače. Během tohoto pohybu, ale musí uživatel poslat aplikaci vstup s významem upuštění předmětu. Generovat tento vstup stiskem tlačítka však není vhodné, neboť při skutečném hodu člověk uvolní sevření namísto jeho zpevnění za cílem stisknutí tlačítka. Proto bylo rozhodnuto, že bude hod uskutečněn naopak uvolněním tlačítka, což je daleko přirozenější varianta. Srovnáme-li za tímto účelem vstupy ovladače *grip* a spoušť, pak je ten druhý jmenovaný mnohem vhodnější (viz kapitola 6). Granát je v okamžiku stisku spouště za účelem hodu odjištěn a exploduje po třech vteřinách. K tomu dojde bez ohledu na to, zda byl granát vržen či nikoliv.

4.2.4 Další objekty

Do scény měly být umístěny krabice několika velikostí a hmotností, se kterými by uživatel mohl interagovat. Aby však scéna nepůsobila příliš staticky, byl navržen robot procházející se po scéně. Jeho proporce a stavba jeho těla jsou shodné s lidskými, aby bylo možné jeho animace použít pro jiné modely stejného typu, nebo naopak aplikovat jeho animace na podobné modely. Chodící robot sice statickou scénu ožíví, ale pokud by s ním nebyla možná interakce, působil by značně rušivě. Proto je umožněno jej poškodit či zničit pomocí předmětů, které má uživatel k dispozici.

5 Implementace

Pro implementaci navržené aplikace byl zvolen engine Unity. Ten má i ve své bezplatné verzi dostupné všechny části, které jsou pro vývoj téměř libovolné aplikace třeba. Unity obsahuje mocný a pohodlný editor scén, které jsou tvořeny pomocí takzvaných **prefabů**. Jedná se o objekty, které si programátor vytvoří z modelů, jednotlivých definovaných komponent a skriptů v jazyce C# nebo JavaScript (pro tuto práci byl zvolen jazyk C#).

Pro účely zpracování vstupů HTC Vive byly použity dvě knihovny, a sice poměrně nízkourovňová SteamVR a nadstavba nad ní VRTK (Virtual Reality ToolKit). Tyto knihovny umožnily při programování odchylovat události hardwaru a vstupy uživatele, stejně jako ovlivňovat jeho činnost jako například setměním obrazu, nebo vibrací ovladače.

Implementace aplikace byla provedena v souladu s návrhem, a proto se tato kapitola věnuje pouze upřesněním, odchylkám nebo některým implementačním detailům, které v kapitole věnující se návrhu nejsou probrány.

5.1 Implementace chování

Cílem této podkapitoly je přiblížit programové pozadí a skripty, které byly naprogramovány pro vstupy uživatele, jeho interakce s prostředím a ostatní dění ve scéně.

5.1.1 Uživatel

Pro implementaci přítomnosti uživatele ve scéně byl v základu využit prefab CameraRig knihovny VRTK. Ten v sobě obsahuje zkalibrované hranice místnosti, snímání pozice hardwaru a realistické modely ovladačů pro jejich zobrazení ve scéně. Jeho funkčnost však bylo nutné pro účely tvořené aplikace rozšířit.

Do prefabu byl přidán skript pro **ztmavení** displeje HMD v případě, že se náhlavní display nachází v objektu ve virtuální scéně. Byl však modifikován, aby nedošlo ke kompletnímu ztmavení, ale uživatel stále viděl dostatečně na to, aby si uvědomoval svou polohu a své pohyby. Tím byla odstraněna dezorientace a v některých případech pocit strachu při aktivaci skriptu. K tomu totiž nemuselo dojít, pouze pokud uživatel úmyslně strčil hlavu do zdi, ale také v případě, že na něj objekt spadl, nebo do něj, v případě robota, objekt vešel.

Dále byl využit skript pro implementaci **teleportu**. I ten však musel být modifikován. Jak se dočteme níže, uživatel má v moci zpomalit běh času ve scéně. To však mělo za následek i zpomalení teleportace. V kapitolách návrhu a testování je naznačeno, že teleport formou rychlého přesunu testerům nedělal problémy, avšak v případě zpomaleného času a tedy i tohoto přesunu se již často dostavovaly nepříjemné pocity. Proto byl skript uskutečňující teleportaci upraven, aby bral v potaz změněný běh času. Teleport je po jeho výběru aktivován stiskem spouště. Zatímco je stisknutá, vykresluje se křivka, na jejímž konci určeném kolizí s objektem ve scéně se zobrazí značka reprezentující místo, kde bude uživatel stát po teleportaci uskutečněné uvolněním tlačítka. Oblast mimo vytvořenou scénu byla pro skript teleportu označena tak, aby mimo ni nebylo možné vkročit.

Další skript umožní **restartování scény** například pro případ, že se v ní nachází obrovské množství vytvořených objektů, které uživatel nechce uklízet, avšak chce pokračovat v používání aplikace. Restartování je docíleno stisknutím všech tří hlavních vstupů ovladače (spoušť, touchpad a grip) současně.

Pro každý z ovladačů byl vytvořen skript umožňující přístup do **kruhového menu** popsaného v kapitole 4.1.2. Skript při dotyku palce touchpadu zobrazí menu o čtyřech segmentech založené

na implementaci knihovnou VRTK. Horní tři segmenty reprezentují kategorie předmětů (zprava chladné, střelné a vrhací zbraně) a spodní segment zpřístupňuje teleport. Položka je vybrána stisknutím touchpadu, zatímco je položka zvýrazněna (neboli zatímco je jí palec na touchpadu nejbližší).

Výběr kategorie zbraní sice žádnou uživateli nevloží do ruky, ale původní kruhové menu je nahrazeno novým, reprezentujícím danou kategorii. Výběr položky v tomto menu již předmět zpřístupní a v případě výběru položky s ikonkou rudé šipky dojde ke skrytí současného zanořeného menu a zpřístupnění menu hlavního kruhového s kategoriemi a teleportem. Menu je vykreslováno speciálním shaderem, který umožní jeho vykreslení i skrz držené předměty.

Každý ovladač disponuje vlastním nezávislým kruhovým menu s vlastním kontextem. To umožní uživateli například mít v každé ruce jiný (nebo v obou typově ten stejný) předmět, nebo mít v levé ruce aktivovaný teleport, s pomocí něhož se pohybuje po scéně, a v druhé libovolný předmět.

5.1.2 Předměty

V okamžik, kdy je některý předmět prostřednictvím kruhového menu vybrán poprvé, je uživateli čerstvě vytvořen přímo do ruky. V případě, že uživatel vybere jiný předmět nebo teleport, ten současný je deaktivován, skryt a uložen, aby mohl být opět aktivován, jakmile bude znovu vybrán. To umožní například vyprázdnit polovinu zásobníku pistole, vybrat jiný předmět a po opětovném výběru pistole na tomtéž ovladači bude mít zbraň v zásobníku stále stejný počet patron.

Chladným zbráním ale i ostatním předmětům s výjimkou teleportu jsou definovány kolizní zóny pro účely fyzikální interakce s ostatními objekty ve scéně. Využití simulačního fyzikálního enginu vestavěného v Unity je však zkomplikováno tím, že držený předmět je ve smyslu hierarchie stromu objektů ve scéně učiněn potomkem ovladače a dokonale sleduje jeho pohyb. Toto řešení bylo zvoleno, abychom se vyhnuli nepříjemné situaci, kdy bychom například drženým předmětem udeřili do virtuální zdi a ačkoli by skrz ni ovladač zákonitě prošel, držený virtuální předmět by jí byl zastaven. Tento přístup však, jak bylo předesláno, komplikuje využití fyzikálního enginu tak, že je nutné v okamžiku kolize s objektem manuálně aplikovat sílu na daný bod zasaženého tělesa. Velikost a směr síly je vypočtena z vektoru rychlosti ovladače a hmotnosti drženého předmětu, přičemž těžší předměty (jako například palcát ve srovnání s katanou) udělí zasaženému objektu větší hybnost.

Střelné zbraně kromě výstřelu, při absolutním stisku spouště ovladače, neustále detekují míru jejího stisku a tuto informaci využívají k pozicování spouště zbraně, takže když uživatel stiskne spoušť pouze do poloviny jejího maxima, může ve virtuální realitě pozorovat, že i spoušť zbraně, kterou drží je napůl zmáčknutá.

Každý výstřel pistole způsobí sérii následujících akcí: Po kontrole, že je v zásobníku dostatečné množství střeliva (v opačném případě bude pouze přehrán zvuk prázdného klapnutí kladívka zbraně) a od posledního výstřelu uplynul dostatečný čas, je přehrán zvuk výstřelu, je zobrazen záblesk výstřelu tvořený plameny, světlem a kouřem, je přehrána animace zpětného rázu výstřelu, v zásobníku je skryt vystřelený náboj, ostatní náboje zásobníku jsou posunuty pružinou blíže ke komoře pistole, je vytvořena a ze zbraně vymrštěna prázdná nábojnice ovlivňovaná fyzikou, je vykreslena dráha letu kulky, na místě jejího dopadu je vytvořen částicový efekt několika jisker a obláček kouře, zasaženému objektu je udělena kinetická energie a v případě, že se jedná o objekt splňující interface (užitečný konstrukt jazyka C#) *Activatable*, které bylo vytvořeno pro tento účel, je objekt aktivován. Tyto objekty budou blíže popsány v následující podkapitole.

Brokovnice funguje obdobně s těmi rozdíly, že jediným stiskem spouště vystřelí šestnáct broků a vzhledem k absenci zásobníku po každém výstřelu není vyhozena nábojnice, ale jsou všechny tři vyhozeny až při zalomení zbraně při přebíhání.

Přebíjení všech střelných zbraní funguje obdobně. Starý zásobník (či nábojnice v případě brokovnice) je oddělen od zbraně, je mu vytvořena komponenta *RigidBody*, která umožní jeho fyzikální

simulaci, avšak kolize s objekty ve scéně je aktivována až o okamžik později, aby nedošlo ke kolizi se zbraní, ze které byl uvolněn. O pár zlomků vteřiny později je vytvořen nový zásobník a formou animace je zasazen do zbraně, které je opět umožněna střelba.

Jednoruční revolverový raketomet se od textu výše odlišuje jen v několika bodech. Předně je to ten, že po výstřelu není z jeho hlavně vržen paprsek imitující let kulky, ale je vytvořena raketa, která za hluku svého motoru poletí rovně ve srovnání s kulkou relativně pomalu, dokud nenarazí do objektu. Jakmile k tomu dojde, je raketa zničena a v bodě nárazu je vytvořena pomyslná koule o poloměru definovaným parametry rakety. Na objekty uvnitř této koule zapůsobí síla směřující vždy od jejího středu a o velikosti závislé na vzdálenosti od něj a na dalším parametru rakety. Na místě dopadu je rovněž vytvořen efekt exploze sestávající z poloprůhledně oranžovožlutě texturovaných objektů, a sice koule s nepravidelným povrchem a několika prstenců, které se velmi rychle zvětšují a dále zprůhledňují různými rychlostmi. Součástí efektu je také bodový zdroj žlutého světla, jehož intenzita v čase klesá úměrně celkové průhlednosti zvětšujících se objektů, a slabý částicový efekt kouře. Jelikož jsou letící rakety objekty ovlivňované fyzikou, je možné, aby například při výstřelu dvou raket rychle za sebou na vzájemně si blízké cíle, výbuch první rakety vychýlil raketu druhou, čímž z její přímé dráhy vytvoří spirálu.

Pistole disponuje holografickými mířidly, které jsou implementovány následujícím způsobem. Před zbraň přibližně nad konec hlavně je do výšky holografického sklíčka umístěn „holografický“ obraz mířidel. Ten je umístěn v objektu, který má zjednodušeně tvar kvádrů s chybějící stěnou v místě sklíčka a normálou všech ostatních stěn směřující dovnitř. Na oba objekty (kvádr i mířidla) je aplikován shader, který umožní, že kvádr je vždy neviditelný a mířidla jsou vidět pouze, když na ně pohlédneme skrz chybějící stěnu kvádrů, tedy holografické sklíčko. Ačkoli by holografická mířidla měla pro maximální přesnost ideálně být v nekonečné vzdálenosti, toto řešení je poměrně dobrou aproximací na malé až nižší střední vzdálenosti, což je konec konců vzdálenost, odpovídající ve skutečné realitě efektivnímu dostřelu pistolí.

Všechny předměty přístupné z kruhového menu dědí ze třídy *IObject* (zkratka pro Interactable Object), která je nadstavbou nad třídou knihovny VRTK. Původní třída umožňuje snadné odchyťávání vstupů ovladačů (které bylo dále rozšířeno použitím komponent/skriptů jako *VRTK_ControllerEvents* nebo *VRTK_ControllerEvents_UnityEvents*) a byla rozšířena například o možnost aktivace objektů úderem, o vylepšení interakce s objekty zahrnutými do fyzikální simulace nebo o metody pro automatické zničení odhozeného objektu či aktivace jeho kolize po definované době.

Objekty jako **vrhací hvězdice** nebo **granáty**, jsou instance třídy dědící z *IObject* a dále jej rozšiřují modifikací metody automaticky zavolané při upuštění objektu. Tato modifikace spočívá v udělení rotace vrženému objektu (neboť se jedná o vrhací hvězdicu a německý granát z druhé světové války a oba tyto předměty jsou touto rotací charakteristické), která vychází ze síly hodu a je pro lepší vizuální efekt shora omezená, a dále v zavolání korutiny *RefreshItem* komponenty ovladače, která byla implementována pro řešení zobrazení a funkčnosti kruhového menu a tedy i pro tvorbu předmětů do ruky uživatele. Tato korutina o okamžik později vytvoří nový předmět stejného typu, jako byl ten vržený. Jak již bylo zmíněno, objekt je vržen uvolněním spouště ovladače. Z toho však plyne, že je nutné spoušť nejprve stisknout. V případě granátů tento stisk od předmětu samotného oddělí pojistku, aktivuje její fyzikální simulaci a odstartuje třívteřinový (myšleno herního a nikoliv reálného času) čítač. Po vypršení této doby je granát zničen a na jeho místě je vytvořen efekt výbuchu spolu s působící silou exploze. Neboť je možné, že granát detonuje ještě v uživatelské ruce, je nutné tuto skutečnost zkontrolovat a v takovém případě ještě zavolat korutinu *RefreshItem*, neboť ta je obvykle volána až v okamžiku hodu výbušniny. Je také ale nutné vyřešit, že pouhé zavolání této korutiny by vedlo na upuštění nově vytvořeného neodjištěného granátu v okamžiku, kdy by uživatel uvolnil spoušť (která je v tomto případě zákonitě stále stisknutá, neboť jsme ve stavu, kdy granát explodoval v ruce uživatele,

k čemuž by nemohlo dojít, kdyby spoušť stisknutá nebyla, neboť by její uvolnění vedlo k upuštění granátu). Je tedy nutné korutinu zavolat až po uvolnění spouště.

Aby bylo možné u pistole a brokovnice použít stejnou funkci pro let kulky, byla vytvořena abstraktní třída *AbstractGun* se statickou metodou *BulletHit*. Tato metoda je volána pro každý výstřel pistole a pro každý brok výstřelu brokovnice.

Další zobecnění se týká granátů, raket a barelů, které budou popsány v příští podkapitole. Všechny tyto objekty jsou výbušniny, a proto bylo vhodné vytvořit jedinou funkci *Explode*, která tento jejich aspekt řeší. Komplikace však nastala v tom, že nebylo možné pro všechny tyto objekty vytvořit jedinou třídu *Explodable*, neboť třída granátů již dědí ze třídy *Thrown* (ve které se nachází vrhací hvězdice) a jazyk C# neumožňuje vícenásobnou dědičnost. Z toho důvodu byla vytvořena statická metoda *StaticExplode*. Rakety a barely tuto metodu mohou volat přímo a granáty ji pouze využívají bez tvorby instance její třídy pomocí předání parametrů ze svých atributů.

Pro aplikaci bylo vytvořeno několik zvukových efektů pro střelbu, exploze a zvuk letu rakety pomocí programu Audacity.

Dodejme ještě, že komponenta ovladače obstarávající funkčnost kruhového menu se nazývá *ItemManager*, a že všechny objekty jako vystřelené nábojnice, vyhozené zásobníky nebo vržené předměty jsou automaticky odstraněny po uplynutí časového intervalu.

5.1.3 Scéna

Programové chování scény je poměrně jednoduché. Všechny objekty v ní umístěné vyjma statických, jako jsou země, mosty nebo sloupy, jsou zahrnuty do fyzikální simulace vestavěného fyzikálního enginu. Všechny tyto objekty mají nastavenou takovou hmotnost, aby je exploze, střelba a úderu uživatele ovlivňovaly ve vhodné míře.

Ve scéně jsou také umístěny objekty, které jsou z fyzikální simulace vynechány, avšak skript, který byl pro ně vytvořen, umožňuje, že při zásahu takového objektu je vytvořena nová instance jiného objektu zpravidla už ovlivněného fyzikálním enginem. Tyto tvořící objekty splňují již zmíněný interface *Activatable* a onou aktivací je v jejich případě chápána právě tvorba nových objektů po zásahu.

Všechny objekty tohoto typu sdílí stejný skript, který lze však silně parametrizovat a měnit tak jeho chování. Skriptu je možné přiřadit jeden nebo více prefabů objektů, které má tvořit a také jeden nebo více objektů ve scéně, které budou reprezentovat pozici, na níž budou objekty vytvářeny. Dále je možné specifikovat počet objektů, které mají být vytvořeny jedinou aktivací (tato hodnota je vzhledem k jednoduchosti opětovné aktivace u většiny objektů nastavena na jedna). V případě, že je definováno více různých předmětů, které mají být tvořeny, při každé tvorbě je z jejich seznamu jeden vybrán náhodně. Dalším parametrem je rychlost, jakou mají nové objekty okamžitě po vytvoření získat. To umožnilo za pomoci stejného skriptu tvorbu kanonů, které tvořené objekty vymrští na velkou vzdálenost. Posledním parametrem je vektor určující rozptyl oproti dopřednému směru objektu definujícího pozici nových objektů, který se aplikuje na objekty tvořené s počáteční rychlostí.

Dalším aktivním prvkem scény splňující interface *Activatable* je helikoptéra. Ta se nachází stále ve stejné pozici a je jí přehrávána stále stejná animace otáčení obou rotorů. Aktivováním helikoptéry, tedy střelbou do ní, je zpomalen herní čas. Toho bylo dosaženo změnou atributu *timeScale* třídy *Time*. Dále je však nutné změnit i hodnotu atributu *fixedDeltaTime* na hodnotu $slowMotionSpeed \times 0.02$, kde *slowMotionSpeed* je nová hodnota atributu *timeScale* určující novou rychlost plynutí času a *0.02* je výchozí hodnota této proměnné. *FixedDeltaTime* je nutné upravit proto, neboť má tato hodnota vliv na výpočty simulačního enginu a tak by pozice a rotace objektů jím simulovaných byly obnovovány s původní četností. Opětovnou aktivací helikoptéry je rychlost času nastavena na původní hodnotu.

Aktivační interface splňuje rovněž výbušný barel. Jeho aktivace spočívá jednoduše v jeho explozi, tedy zničení jeho samotného, tvorbě efektu výbuchu a aplikace odpovídající síly na blízké objekty se silou klesající nepřímo úměrně vzdálenosti od místa detonace a parametrizovanou silou výbuchu.

Posledním prvkem scény se specifickým chováním je robot. Ten obsahuje komponentu *Animator* (podrobněji popsána v kapitole 5.2.3) podobně jako střelné zbraně, avšak graf jeho stavů je velmi jednoduchý, neboť je neustále přehrávána pouze animace jeho chůze. Skript řešící chování robota obsahuje dvě části. První se stará o pohyb během chůze a druhá o oddělování jeho částí.

Rychlost přehrávání animace chůze byla zvolena tak, aby působila pokud možno přirozeně a rychlost samotného pohybu robota scénou na oplátku tak, aby jeho chodidla neklouzala po podlaze vlivem příliš vysoké respektive nízké rychlosti posunu animovaného modelu. Definice trasy byla řešena velmi jednoduše. Skript zkontroluje, zda se v hierarchii objektu robota nachází více potomků, než animační kostra, a pokud ano, předpokládá, že se jedná o značky, kudy se má pohybovat. Při vytvoření robota případně při spuštění scény jsou pozice těchto objektů ve scéně uloženy do vlastního pole, přičemž je jejich pořadí důležité. Následně je započat pohyb robota konstantní parametrizovanou rychlostí směrem k prvnímu definovanému bodu. Jakmile se ocitne v dostatečné blízkosti tohoto bodu (určenou dalším parametrem), vydá se směrem k dalšímu. Tento postup se opakuje, dokud nedorazí k poslednímu bodu své cesty. Následně je zkontrolován booleovský parametr skriptu určující, zda definované body tvoří cyklickou trasu. Pokud ano, pohyb robota pokračuje směrem k prvnímu bodu a pokud ne, robot se otočí a vrací se ve svých stopách na začátek. Pokaždé, když dorazí k dalšímu bodu trasy (případně při tvorbě robota či spuštění scény) je mu vypočtena nová rotace za pomoci funkce *Quaternion.LookRotation(direction)*, kde *direction* je směr k nadcházejícímu bodu, a mezi jeho současnou a cílovou rotací je uskutečněn plynulý přechod pomocí funkce *Quaternion.Lerp(r1, r2, t)*, kde *r1* a *r2* jsou původní respektive cílová rotace a *t* je hodnota v rozmezí 0 až 1 vyjadřující blízkost aktuální rotace té původní respektive cílové.

Druhá část skriptu robota obstarává jeho aktivaci, ke které pro připomenutí dojde při zasažení kulkou, explozí či úderem drženou případně vrženou zbraní. Aktivace má sémantiku oddělení zasažené části těla. Prefab robota a kód tohoto skriptu kvůli tomu bylo nutné implementovat následujícím způsobem. Jednotlivé části robota bylo nutné v hierarchii ručně umístit pod animační kosti, které jim odpovídají a těmto kostem (nikoliv modelům částí těla) byly přiřazeny komponenty *Collider* zajišťující detekci kolizí. Vzhledem k tomu, že tento skript, který splňuje interface *Activatable*, je dobrým návrhem mít připojen pouze k nejvyššímu objektu v hierarchii robota, je při pokusu o aktivaci zasaženého objektu (který obecně nemusí být robot, ale libovolný objekt ve scéně), nutné najít v grafu scény jeho předka, který toto interface splňuje (pokud takový existuje). To je obecně nutné i u ostatních objektů, které lze aktivovat, ale všechny jejich komponenty *Collider* se nenachází v nejvyšší úrovni jejich hierarchie.

Každá část robota je označena jedním z *tagů* (způsob jak v Unity efektivně označit typ objektu) *Robot* nebo *RobotLegs*. Pokud je zasažena část těla s *tagem Robot*, je uložena její pozice a rotace, dále je uložen čas přehrávané animace, zasažené části je odstraněn odkaz na svého předka a komponentě *Animator* zbytku robota je zavolána metoda *Rebind*, která způsobí, že oddělená část těla nebude dále animována (kvůli tomu je však také nutné následně nastavit přehrávanou animaci, jež tím byla restartována, zpátky do dříve uloženého času). Oddělená část těla je dále umístěna se svou uloženou rotací do uložené pozice ve světě a je jí přidána vlastní komponenta skriptu robota (aby bylo možné oddělenou končetinu dále dělit) a komponenta *RigidBody*, aby byla obsažena ve fyzikální simulaci. V případě, že byla zasažena část těla s *tagem RobotLegs*, který, jak název napovídá, naznačuje, že byly zasaženy nohy robota, je postup obdobný. Rozdíl spočívá pouze v tom, že animace chůze je zastavena stejně jako první část skriptu věnující se pohybu robota. Zmíňme ještě, že exploze každou část robota aktivují s pouze třetinovou pravděpodobností, aby nedošlo k jeho kompletnímu rozložení.

5.2 Tvorba modelů, textur a animací

Ačkoli byla tvorba modelů, textur i animací pro tuto aplikaci časově i z tvůrčího hlediska náročná, neboť jsou všechny autorské, text, který se těmito aspekty práce věnuje, bude vzhledem k charakteru provedených úkonů stručný. Jak již bylo zmíněno v kapitole 3.2.1, pro tvorbu všech modelů i animací byl zvolen program Blender. Veškeré textury pak byly tvořeny buď rovněž v Blenderu, nebo pomocí programů pro rastrovou grafiku GIMP či vektorovou grafiku Inkscape. Ukázky vytvořených assetů můžeme vidět v příloze 1.

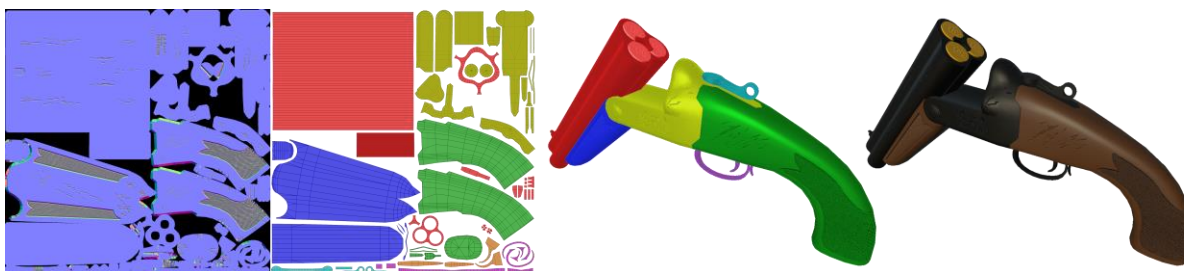
5.2.1 Modely

Pro většinu modelů bylo ještě před započítím modelovacího procesu vytvořeno několik náčrtů vystihujících základní ideu, proporce a design cílového výtvaru. Proces tvorby je totiž značně ulehčen a urychlen, když je možné sledovat podkladový materiál umístěn jako pozadí přímo do modelovacího nástroje. Pokud má modelář pro svou práci k dispozici kvalitní půdorys a bokorys, může být velké množství jeho práce zjednodušeno téměř na jejich obtahování nově tvořenou geometrií. Modely byly tvořeny takovým způsobem, aby reflektovaly požadavky zmíněné v kapitole 4.2 (například, aby bylo množství střeliva v zásobníku viditelné během střelby) a současně aby působily esteticky přívětivě.

Vzhledem k tomu, jakým způsobem funguje v Unity fyzikální engine, bylo pro některé objekty nutné vytvořit pomocné modely. V Unity totiž každý objekt, který má být zahrnut do fyzikální simulace musí ve své hierarchii obsahovat alespoň jednu komponentu typu *Collider*. Tato komponenta definuje tvar, který bude použit pro účely fyzikální simulace. Jedná se pokud možno o co nejjednodušší tvary. Hlavními jsou koule (definovatelná jako vzdálenost od bodu), kapsle (definovatelná jako vzdálenost od úsečky) a kvádr (omezený šesti rovinami). Většina objektů je příliš složitá na to, aby bylo možné je věrohodně popsat jediným takovým objektem a proto je možné těchto *Colliderů* použít více na jeden objekt. Některé modely jsou ale i přes to tvarově příliš složité, případně by bylo příliš pracné jejich tvar pomocí těchto tvarových primitiv dobře aproximovat. Z tohoto důvodu Unity nabízí další možnost popisu tvaru objektů pro účely fyzikální simulace, a sice takzvaný *MeshCollider*. Jedná se o *Collider* jehož tvar je definovaný polygonálním modelem dodaným vývojářem. Ačkoli je výpočet kolize objektů s touto komponentou náročnější než s ostatními primitivy, nepředstavuje při rozumném užití a návrhu aplikace zbytečné komplikace.

Je však důležité být si vědom limitů tohoto nástroje. Pokud použijeme komplexní *MeshCollider* nekonvexního tvaru (což znamená, že existuje spojnice dvou bodů uvnitř tělesa, která není celá obsažena v tomto tělese), můžeme očekávat, že například malé a rychle se pohybující objekty nezachytí vždy s takovým *MeshColliderem* kolizi ani v případě, že pro tento objekt aktivujeme její spojitou detekci (neboli, že bude kontrolována i kolize, k níž by došlo mezi dvěma vykreslenými snímky). Z tohoto důvodu bylo pro některé objekty (zejména skály, mosty a věže v mapě), které bylo obtížné aproximovat zmíněnými primitivy, nutné vytvořit pomocné polygonální modely. Tyto modely byly v podstatě rozřezané původní modely zjednodušené tak, aby aproximovaly původní tvar. Tyto modely pak ve scéně nebyly vykreslovány, ale byly umístěny na stejné místo jako model, jež aproximují, a byla pro ně aktivována detekce kolize pomocí *MeshCollideru*, u kterého byla aktivována funkce tvorby nejjednoduššího konvexního tvaru z dodaného modelu.

Pokud je povrch vytvořeného modelu vhodné popsat více než jedním materiálem, je nutné po dokončení fáze modelování vytvořit jednotlivé materiály a označit stěny, které k nim patří. Tímto způsobem lze zařídit, aby například čepel japonského meče byla lesklá, ale stuha omotaná kolem jeho jílce matná. Pokud navíc chceme využít obrázek jako texturu pro model, musíme definovat mapování trojrozměrného objektu do roviny textury. Nejuniverzálnější způsob tohoto mapování se označuje jako



Obrázek 13: Ilustrace UV mapování textur

UV mapování (pojmenované po osách popisujících pozici v textuře). Idea tohoto způsobu převedení povrchu trojrozměrného objektu do dvou dimenzí je v podstatě ekvivalentní rozřezání jeho povrchu na díly, které by byly rozprostřeny do roviny a následně položeny na cílovou texturu. Příklad tohoto mapování můžeme vidět na obrázku (Obrázek 13). Je na něm zleva zobrazena normálová mapa (popsaná v další podkapitole), znázorněná UV mapa, UV mapa projektovaná na model a model samotný. Blender umožňuje tvorbu UV mapy několika způsoby. Jedním z nich je funkce *Smart UV project*, která provede automaticky projekci do UV souřadnic na základě pokřivení povrchu a úhlů, které mezi sebou svírají stěny modelu. Alternativní způsob, který umožňuje větší kontrolu nad výsledkem projekce a který byl použit pro účely tohoto projektu, je manuální označení hran modelu po kterých chceme, aby byl proveden pomyslný řez povrchu za účelem jeho vyhlazení do roviny. Po označení všech hran je možné zavolat funkci *Unwrap*, která provede projekci do roviny a výsledné mapování je ještě možné manuálně upravit. Pro maximální kontrolu je možné provést projekci několika objektů do jedné textury, nebo naopak mít více mapování pro různé části jednoho modelu.

Ačkoli ve výsledné aplikaci není následující asset trojrozměrným modelem, přesto pro něj byl model vytvořen. Řeč je o takzvaném **skyboxu**, který definuje, co má být uživateli zobrazeno v místech, kde pomyslný paprsek vycházející z kamery nedopadne na žádný objekt ve scéně. Jak už název napovídá, skyboxy se užívají nejčastěji pro popis oblohy a je možné je popsat jako nekonečně velkou kostku, ve které je scéna umístěna. Nekonečná velikost kostky v podstatě znamená, že to co na skyboxu uživatel uvidí je nezávislé na jeho poloze, ale pouze na směru pohledu.

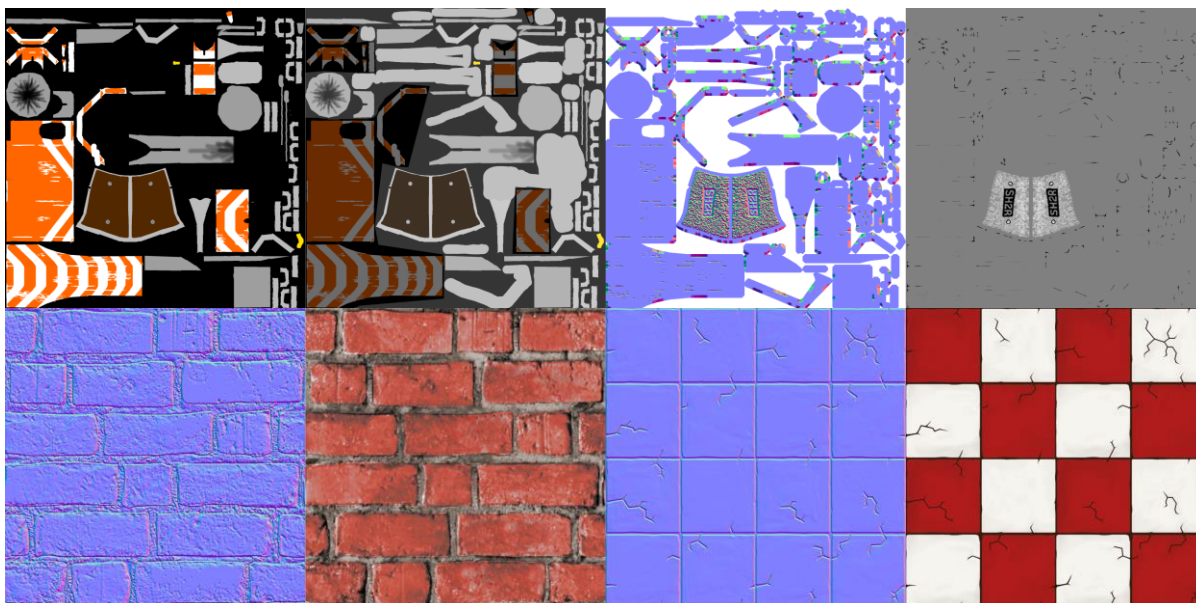
Skybox byl vytvořen tak, že byla v Blenderu vymodelována (a na tento model nanášena později textura) scéna hor a nebe s mraky a do jejího středu bylo postaveno šest kamer vzájemně otočených o 90 ° se zorným úhlem o stejné hodnotě, každá z nich reprezentující jednu ze stěn kostky a v součtu celou krychli dokonale popisující. Následně bylo provedeno vykreslení scény každou z kamer, výsledek uložen a z něj v Unity vytvořen materiál skyboxu.

5.2.2 Textury

Většina textur byla vytvářena jedním ze dvou způsobů. Prvním je vytvoření UV mapy, jejího exportu a její následné otevření v programu GIMP či Inkscape, kde jsou do polygonální sítě „rozřezaného“ modelu vkresleny detaily.

Druhým způsobem je po vytvoření UV mapy využít nástroje Blenderu, které umožňují do různých slotů textur kreslit přímo na trojrozměrný model. Pokud se tvůrce textury sžije s ovládáním těchto nástrojů a nastavením jejich štětců, jedná se diskutabilně o nejpohodlnější způsob, jak texturu vytvořit.

Ke tvorbě textur byly dále použity dva odlišné přístupy ještě v jednom smyslu, a sice v původu výchozích obrázků. Některé textury byly tvořeny od základu ručním kreslením, ale pro některé byly jako podklad použity fotografie pořízené za tímto účelem. V obrázku (Obrázek 14) můžeme ve spodním řádku vidět srovnání textury kompletně nakreslené ručně (popraskané dlaždice) a vytvořené z fotografie (cihlová zeď). Druhý jmenovaný případ byl použit zejména pro tvorbu textur mapy, jako



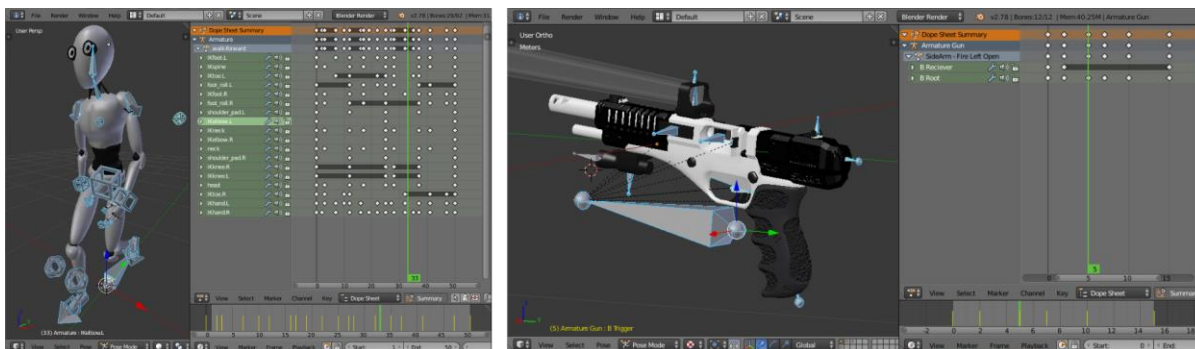
Obrázek 14: Příklad různých typů vytvořených textur (raketomet, stěna a podlaha)

je například zmíněná cihlová zeď či travnatá zem. Za tímto účelem musely být použité fotografie silně upraveny. Byly v nich odstraněny stíny, neboť ty mají být produktem světla a geometrie (případně textur, jak bude brzy rozebráno) ve scéně, a byly upraveny tak, aby na sebe navazovaly. Posledně jmenovaný aspekt je myšlen tak, že při naskládání textury těsně vedle sebe v obou osách nikdy není vidět předěl mezi dvěma obrázky. Tato vlastnost je výhodná právě v námi požadovaném případě, kdy například stěna je mnohem větší než použitá textura a tak dojde k jejímu naskládání vedle sebe. Této vlastnosti bylo dosaženo tak, že byla textura posunuta v obou osách o polovinu svých rozměrů (přičemž přesahující textura byla vždy vložena na opačnou stranu), čímž se předěly dostaly do středu obrazu, a ty byly následně vyretušovány.

Jak bylo naznačeno výše, textura nemusí definovat pouze barvu povrchu modelu. Lze ji také využít například pro modifikaci tvaru povrchu, který je na modelu původně rovný, pro účely odrazu světla. Tento efekt je velmi užitečný pro vytvoření iluze nerovností povrchu, který má ve skutečnosti mnohem méně komplexní geometrii. Lze toho dosáhnout použitím jednoho ze dvou speciálních typů textur, konkrétně bump mapy a normálové mapy.

Bump mapa, popisuje odchylky od původního povrchu modelu černobílou texturou. Padesátiprocentní šedá barva vyjadřuje, že je výška povrchu beze změny, světlejší barva vyjadřuje zvýšený povrch a tmavší naopak snížený. Při odrazu světla od povrchu modelu modifikovaného bump mapou je spočtena nová normála povrchu na základě intenzity čtyř sousedících pixelů bump mapy v místě dopadu paprsku. Výhodou této textury je, že je poměrně snadné ji vyrobit. Příklad bump mapy můžeme vidět na obrázku (Obrázek 14) v první řadě úplně vpravo (zleva textura popisující barvu, spekulární textura, normálová mapa a bump mapa). V textuře jsou viditelné zakreslené šrámy na zbrani černou barvou a výlis textu „SH2R“ na pažbičce.

Normálová mapa je funkčně v podstatě totožná s bump mapou. Rozdíl spočívá v tom, že normála povrchu, která se vypočte s pomocí bump mapy je v normálové mapě přímo uložená a to v podobě trojrozměrného vektoru, kdy je hodnota každé jeho dimenze uložena do jednoho z barevných kanálů textury. Jedná se tedy o optimálnější řešení, avšak její výroba je složitější. Naštěstí Blender disponuje nástroji pro konverzi bump map do mapy normálové a tak byly všechny textury modifikující vlastnosti povrchu tohoto typu v tomto projektu řešeny pomocí normálových map. Příklad mapy vytvořené z bump mapy můžeme vidět na obrázku (Obrázek 14) v horním řádku.



Obrázek 15: Ilustrace průběhu tvorby animací

Posledním zvláštním typem textur, které byly v této práci použity a tvořeny jsou takzvané **speculární mapy**. Tyto textury definují míru odrazivosti povrchu objektu. Díky nim můžeme jedinou texturou popsat celý objekt, ačkoli obsahuje velmi odlišné materiály. Na příkladu raketometu z obrázku (Obrázek 14) můžeme vidět, že lesklé kovové části jsou ve stejné textuře jako méně lesklá plastová rukojeť zbraně, a rozdílné vlastnosti těchto povrchů jsou vyjádřeny ve speculární textuře (druhá zleva) odlišnou barvou a jejím jasnem. Na velmi lesklých částech modelů můžeme navíc v běžící aplikaci vidět odraz skyboxu, což dává další důraz na jeho kvalitní vypracování.

5.2.3 Animace

Po dokončení tvorby modelu je v případě, že má být model animován kosterní animací, nutné vytvořit onu kostru (*Armaturu*) vytvořeného objektu. V případě robota je základní vytvořená kostra velmi podobná kostře lidské, avšak pro pohodlnou práci při tvorbě animací je vhodné armaturu značně rozšířit. Jednotlivým kloubům je možné definovat limity, do kterých jim bude dovoleno se ohýbat, ale diskutabilně nejužitečnějším nástrojem, který je možné pro snadnější animaci vytvořit je použití **inverzní kinematiky (IK)**.

Základní animace se provádí postupným umísťováním a rotací požadovaných kostí v daném čase animace do cílové polohy a vložení pro ně klíčových snímků do časové osy (Obrázek 15). Je tak zpravidla konáno v hierarchii kostí shora dolů, tedy například od ramene k zápěstí (nejprve rotujeme celou paži, poté předloktí a pak zápěstí). Právě v případě tvorby animace ruky by však většinou byl pohodlnější opačný postup, kdy bychom umístili zápěstí do požadovaného místa a předloktí a paže by samy rotovaly tak, aby bylo zápěstí v námi určené poloze. Tento postup je možný a nazývá se právě inverzní kinematika (neboť hýbeme kostmi v opačném pořadí v hierarchii odspoda nahoru).

Pro tvorbu kostry s inverzní kinematikou končetin jsou použity další kosti reprezentující cílové pozice jednotlivých končetin, přičemž každému IK řetězci je definována délka vyjadřující počet kostí, jejichž transformace má být směrem vzhůru v hierarchii kostí upravována. Navíc je vhodné použít další pomocnou kost, jejíž účel v příkladu animace ruky je ten, že určuje, kam má směřovat loket, neboť obecné řešení inverzní kinematiky mívá nekonečné množství řešení.

Pro účely této práce byla vytvořena velmi komplexní kostra, která kromě popsaných pomocných principů řeší i ohyb páteře a diskutabilně nejsložitější pomocný systém, kterým je chodilo, avšak pro komplexnost tvorby a nastavení výsledné kostry zde nebudeme zacházet do takových podrobností.

Po vytvoření kostry následuje obvykle fáze označována jako *WeightPainting* nebo společně s tvorbou kostry *Rigging*. Tato fáze jednotlivým vertexům definuje, jakou mírou má být během animace jejich pozice ovlivněna pohybem té které kosti. Ačkoli první verze modelu robota touto fází prošla, nakonec byl zvolen jiný přístup pro transparentnější řešení jeho dělení na části uživatelem v aplikaci. Tím přístupem je rozdělení částí modelu robota do jednotlivých objektů, které mohou být odděleny, a vytvoření vztahu otec-syn mezi každou částí těla a její odpovídající kostí.

Pro přehrávání animací byla v Unity použita komponenta *Animator*, která pro svou funkci využívá graf animačních stavů. Stavy samotné jsou importovány ze souboru obsahujícího model a animace. Mezi jednotlivými stavy jsou vytvořeny přechody, které mohou definovat plynulý přechod mezi animacemi a které mohou být aktivovány dokončením přehrávání předchozí animace či změnou parametrů zaslaných této komponentě. Přehrávání stavu je kromě toho možné aktivovat rovněž programově. *Animator* má navíc více vrstev, ve kterých lze mezi stavy přecházet nezávisle a výsledné animace kombinovat. V souboru obsahujícím model navíc lze v každé animaci vytvořit v libovolném čase událost, které může být přiřazena funkce, jež má být v komponentách animovaného objektu nalezena a zavolána jakmile je v této animaci dosažen specifikovaný čas.

Využitím tohoto systému bylo možné například brokovnici animovat následujícím způsobem: Ve vlastní animační vrstvě je pomocí skriptu řešena animace stisku spouště, aby přesně odpovídala stisku spouště ovladače. V další vrstvě je graf stavů, kde je v době nečinnosti aktivní prázdný stav. Při výstřelu skript aktivuje stav pro animaci zpětného rázu. Ve stejném grafu se nachází posloupnost stavů pro přebití zbraně. V okamžiku stisku přebíjecího tlačítka je programově znemožněna další střelba a započne přehrávání animace zalomení hlavní. Na konci této animace ve vytvořena událost, která automaticky zavolá funkci, jež nábojnicím aktuálně nabitým ve zbrani přidá komponentu *RigidBody* pro umožnění fyzikální simulace a udělí jim kinetickou energii, čímž je vymrští z hlavní. Je také spuštěna korutina pro aktivování kolize nábojnic teprve po pár zlomcích vteřiny, aby nekolidovaly už se zbraní, z níž byly vyhozeny. O okamžik později (abychom dali prostor vylétávajícím nábojnicím) jsou vytvořeny nové náboje a je spuštěna animace jejich zasunutí do hlavní. Po jejím dokončení v graf animací automaticky přejde do stavu, který zalomené hlavní vrátí do původní polohy a na konci této animace je pomocí definované události opět volána další funkce signalizující dokončení přebíjení a povolení střelby.

6 Testování

Při práci na tomto projektu byl zvolen přístup častých testů již od počátku vývoje, aby byly problémy v implementaci odchyceny už v zárodku. Tento přístup byl důležitý, vzhledem k potenciálu virtuální reality způsobovat nepříjemné pocity, když je implementována nevhodným způsobem. Navíc časté iterace vedly k mnoha vylepšením a zvýšení uživatelského komfortu. Zbytek této kapitoly je věnován několika zjištěním, které byly při testování objeveny, či které byly na základě testů upraveny. Skupina testerů se skládala z poměrně rozmanitého vzorku lidí, kde byli kromě uživatelů s mnoha zkušenostmi ve VR i úplní nováčci v tomto pohledu.

V konečné aplikaci byly v kruhovém menu pro tvorbu objektů použity obrázky znázorňující miniatury předmětů, které se zobrazí po stisku tlačítka. Těmito miniaturami byly rendery 3D modelů a jejich účel byl usnadnit orientaci v menu při pouhém přejetí očima přes ovladač. Když dále vezmeme v potaz, že velikost menu byla volena tak, aby poměrově k ovladači působila esteticky dobře, není už tak překvapivé, že někteří uživatelé při testování luštili význam obrázků z párcementové vzdálenosti. Jedná se však o problém, který se objevil jen u některých testerů a to ještě pouze ve fázi, kdy nebyli obeznámeni s předměty, které jsou skrz menu přístupné. Jakmile tyto předměty už alespoň jednou viděli, ikona pro ně byla okamžitě rozeznatelná a už plnila svůj účel vhodně. Tento problém by bylo možné v budoucí úpravě aplikace vyřešit použitím velmi jednoduchých piktogramů objektů používajících pouze dvě kontrastní barvy jako černá a bílá namísto skutečných renderů objektů.

Během testování se ukázalo, že některé předměty vytvořené pomocí kruhového menu jsou v ruce uloženy tak, že část menu skrývají. Použití menu pak bylo touto skutečností zkomplikováno. Z toho důvodu byl pro vykreslení menu zvolen odlišný shader, který využívá i knihovna VRTK, jež umožnil vykreslení menu skrz držené předměty.

Velké množství testovacích uživatelů našlo zálibu v opakované střelbě z pistole, a tak byl implementován automatický režim střelby, kdy stačí držet spoušť pro opakovaný výstřel. Tento krok zvýšil uživatelský komfort a byl smysluplný díky dostatečné kapacitě zásobníku (konkrétně o dvaceti ranách). Možnost poloautomatické střelby zůstala zachována v podobě možnosti na ni přepnout pomocí aplikačního tlačítka umístěného nad touchpadem ovladače. Toto je však jediné využití tohoto tlačítka v celé aplikaci, nicméně pro slabý zájem testerů o tuto funkci příliš nevádí její přehlédnutí uživatelem.

Přesnost holografických mířidel se v aktuální implementaci ukázala na velké vzdálenosti jako nedostatečná. Tuto přesnost by bylo možné zvýšit posunutím holografického obrazu mířidel dál od zbraně, ale to by se současným nastavením fronty vykreslovacího řetězce znamenalo objevení nevhodné chyby, kdy by se při míření proti stěně a velkému přiblížení k ní hologram do ní schoval. Alternativou, která by dávala lepší výsledky a kterou navrhuji do dalších rozšíření aplikace, by bylo hologramem emulovat železná mířidla. Konkrétně tak, že přední mířidla by byla reprezentována tečkou ve vzdálenosti, kde se nachází hologram v aktuální implementaci, avšak přibyl by kruhový hologram pro zadní mířidla, který by byl zobrazen v podstatě přímo na sklíčku. Přesná střelba by pak byla možná zarovnáním holografické tečky doprostřed kroužku na sklíčku.

Při střelbě na větší vzdálenosti z pistole některým uživatelům silně klesla přesnost kromě obtížnější koordinaci pohybů také z toho důvodu, že se při každém výstřelu přehrává animace zpětného rázu zbraně a další výstřel se může uskutečnit (a při automatické střelbě také uskuteční) dříve, než hlaveň klesne do své původní polohy. Další rány tedy letí o něco výše, než kam bylo původně zacíleno. Na tuto skutečnost však bylo pohlíženo jako na prvek prohlubující interakci s prostředím a proto byl ponechán. Pokud by se však setkal se silnou nevolí uživatelů, je možné jej odstranit buď zrychlením animace zpětného rázu, aby se zbraň vrátila do původní polohy, nebo prodloužením intervalu, než je zbrani dovoleno vystřelit znovu (případně kombinací obou možností).

Jak již je zmíněno v kapitole věnující se návrhu aplikace, simulovat ve virtuální realitě hod je vhodné uvolněním tlačítka na rozdíl od jeho stisku. Výsledky testování nasvědčovaly, že za tímto účelem je nejvhodnější z možných vstupů zvolit spoušť ovladače. Při této volbě dávali testéři při hodech nejlepší výsledky a neměli strach z upuštění ovladače, což byla obava některých v případě použití tlačítka *grip*, neboť za účelem jeho uvolnění je nutné povolit sevření prstů ruky, které drží ovladač.

Testy se soustředily i na optimalizaci implementace teleportu. Jak již bylo zmíněno výše, setmění obrazovky při přesunu sice nezpůsobovalo nevolnosti, ale zato mělo občas za následek mírnou dezorientaci. Ta byla odstraněna, když teleportace proběhla formou velmi rychlého přesunu bez setmění pohledu, ale malá skupina testerů si na tento pohyb musela pár přesunů zvykat. Testy ukázaly, že pomalý pohyb na cílové místo způsobuje nevolnost a tak umožnily najít optimální nastavení rychlosti a zrychlení při teleportaci. Dále byl díky těmto poznatkům upraven kód knihovny funkce uskutečňující teleport tak, aby ve zpomaleném čase probíhal stejnou rychlostí (ve smyslu byl závislý na běhu reálného času a ne toho ve virtuální scéně).

Se zpomaleným časem souvisí i druhá modifikace knihovny funkce, tentokrát té pro zobrazení kruhového menu. I zde bylo nutné upravit kód tak, aby bral v potaz změnu časového měřítka. Samozřejmostí pak už je, že skript definující chování robota se zpomaleným časem počítá také a robot kráčí rychlostí, na rozdíl dvou zmíněných uživatelských akcí, závislou na virtuálním čase.

Posledním zjištěním týkajícím se měřítka času je, že původní zpomalení na desetinu normální rychlosti uživatelé hodnotili jako příliš velké, a v současné verzi aplikace se proto čas zpomaluje pouze na třetinu.

Znakem poměrně kvalitní imerze ve virtuálním prostředí je, že při ohrožení uživatele virtuálním objektem (například když na něj objekt padá) měli testéři tendence předmětu uhýbat.

Aplikaci lze rozšiřovat nejen přidáváním dalších předmětů, prostřednictvím kterých uživatel může interagovat s okolním prostředím, ale také rozšířením tohoto prostředí ať už ve smyslu zvětšení aktuální scény či přidáním scén nových nebo alespoň dalších prostředků pro interakci uživatele.

7 Závěr

V kapitolách výše byla probrána definice a historie konceptu virtuální reality. Dále byly prozkoumány nejmodernější technologie a zařízení, která jsou vyvíjena buď za účelem použití ve VR, či která se v ní navzdory svému původnímu účelu mohou použít. Zmíněny byly též možnosti reprezentace trojrozměrných modelů tvořících virtuální prostředí a bylo vytvořeno srovnání nástrojů, které lze použít pro jejich tvorbu.

V další kapitole byl podrobně popsán návrh tvořené aplikace, přičemž jsme se věnovali uživatelským interakcím a rozhraní, návrhu předmětů, které bude mít uživatel k dispozici, scény ve které se bude pohybovat a dalších objektů, jež se v ní mají vyskytovat.

Dále jsme se věnovali implementačním detailům a pozadí programové stránky věci a aplikační logiky, ale také tvorbě samotných modelů, animací a několika typů textur popisující vlastnosti jejich povrchu a integraci těchto assetů do tvořené aplikace, která virtuální realitu zpřístupňuje pomocí náhlavního displeje společnosti HTC a jeho ovladačů. Poslední kapitola se věnuje výsledkům a poznatkům, které přineslo testování aplikace.

Bylo dosaženo stanovených cílů a výsledkem praktické části této práce je tedy aplikace ve virtuální realitě pro HTC Vive a sada assetů, které mohou být použity i v dalších projektech. Veškerý obsah aplikace je původní a byly použity pouze výše zmíněné knihovny související s integrací náhlavního displeje a ovladačů do aplikace. Jak je naznačeno v kapitolách výše, práci je také možné rozšířit o další obsah a funkčnost například do podoby produktu schopného úspěšné existence na trhu zábavního průmyslu nebo pro řadu jiných účelů.

Literatura

1. MURPHY, D. Augmented Reality Glasses. In: *Tom's Guide* [online]. 6. října. 2015 [cit. 2016-prosince-29]. Dostupné z: <http://www.tomsguide.com/us/best-ar-glasses,review-2804.html>
2. BURDEA, G. C. a COIFFET, P. *Virtual Reality Technology*. druhá edice. Hoboken (New Jersey, United States): John Wiley & Sons, Inc. 2003. ISBN 0-471-36089-9.
3. AZUMA, R. et al. Recent Advances in Augmented Reality. *IEEE Computer Graphics and Applications*. IEEE Computer Society Press, 2001, 21 (6), 34-47. ISSN 0272-1716.
4. How did virtual reality begin? In: *Virtual Reality Society* [online]. [cit. 2016-prosince-29]. Dostupné z: <http://www.vrs.org.uk/virtual-reality/beginning.html>
5. MAZURYK, T. a GERVAUTZ, M. *Virtual Reality: History, Applications, Technology and Future*. Vienna University of Technology: Institute of Computer Graphics and Algorithms, 1996.
6. HEILIG, M. *Sensorama Simulator*. Spojené státy Americké. Vynález, 3050870. 1962-srpna-28.
7. HEILIG, M. *Stereoscopic-Television Apparatus for Individual Use*. Spojené státy Americké. Vynález, 2955156. 1960-srpna-04.
8. HEILIG, M. *Experience Theatre*. Spojené státy Americké. Vynález, 3469837. 1969-září-30.
9. Rift Overview. *Oculus* [online]. [cit. 2017-ledna-03]. Dostupné z: <https://www3.oculus.com/en-us/rift/>
10. KUMPARAK, G. A Brief History of Oculus. In: *TechCrunch* [online]. 26. března. 2014 [cit. 2017-ledna-03]. Dostupné z: <https://techcrunch.com/2014/03/26/a-brief-history-of-oculus/>
11. Spec Comparison. In: *Digital Trends* [online]. 16. října. 2016 [cit. 2017-ledna-03]. Dostupné z: <http://www.digitaltrends.com/virtual-reality/oculus-rift-vs-htc-vive/>
12. SOUPPOURIS, A. How HTC and Valve built the Vive. In: *Engadget* [online]. 18. března. 2016 [cit. 2017-ledna-04]. Dostupné z: <https://www.engadget.com/2016/03/18/htc-vive-an-oral-history/>
13. LANG, B. Analysis of Valve's 'Lighthouse' Tracking System Reveals Accuracy. In: *Road to VR* [online]. 17. července. 2016 [cit. 2017-ledna-04]. Dostupné z: <http://www.roadtovr.com/analysis-of-valves-lighthouse-tracking-system-reveals-accuracy/>
14. Hardware. *HTC Vive* [online]. [cit. 2017-ledna-03]. Dostupné z: <http://www.vive.com/eu/product/>
15. HUNT, C. Field of view face-off: Rift vs Vive vs Gear VR vs PSVR. In: *VR Heads* [online]. 26. června. 2016 [cit. 2017-ledna-03]. Dostupné z: <http://www.vrheads.com/field-view-faceoff-rift-vs-vive-vs-gear-vr-vs-psvr>
16. PSVR: Technical Specification. *Playstation* [online]. [cit. 2017-ledna-03]. Dostupné z: <https://www.playstation.com/cs-cz/explore/playstation-vr/tech-specs/>
17. QAISER, A. Razer Reveals OSVR – An Open-Source Virtual Reality Headset. In: *Virtual Reality Times* [online]. 8. ledna. 2015 [cit. 2017-ledna-03]. Dostupné z: <http://www.virtualrealitytimes.com/2015/01/08/razer-reveals-osvr-an-open-source-virtual-reality-headset/>
18. What is OSVR. In: *OSVR* [online]. [cit. 2017-leden-3.]. Dostupné z: <http://www.osvr.org/what-is-osvr.html>
19. Tech Specs. *Fove* [online]. [cit. 2017-ledna-03]. Dostupné z: <https://www.getfove.com>

20. Tech Specs. *StarVR* [online]. [cit. 2017-ledna-04]. Dostupné z: <http://www.starvr.com/#techspecs>
21. I/O 2014. *Google* [online]. [cit. 2017-ledna-04]. Dostupné z: <https://www.google.com/events/io/schedule>
22. I/O 2016. *Google* [online]. [cit. 2017-ledna-04]. Dostupné z: <https://events.google.com/io2016/schedule#day2>
23. GearVR. *Samsung* [online]. [cit. 2017-ledna-04]. Dostupné z: <http://www.samsung.com/global/galaxy/gear-vr/>
24. TALABA, D. a AMDITIS, A. *Product Engineering: Tools and Methods Based on Virtual Reality*. Springer, 2008. ISBN: 978-1-4020-8200-9.
25. Phantom Premium Overview. *Geomagic* [online]. [cit. 2017-ledna-04]. Dostupné z: <http://www.geomagic.com/en/products/phantom-premium/overview>
26. LANG, B. Leap Motion Launches Overhauled Hand Tracking Engine That's Made for VR. In: *Road to VR* [online]. 17. února. 2016 [cit. 2017-ledna-05]. Dostupné z: <http://www.roadtovr.com/leap-motion-launches-overhauled-hand-tracking-engine-thats-made-for-vr/>
27. PORTER, J. Could Microsoft's Kinect be reborn as a virtual reality controller? In: *Techradar* [online]. 29. června. 2016 [cit. 2017-ledna-05]. Dostupné z: <http://www.techradar.com/news/gaming/could-microsoft-s-kinect-be-reborn-as-a-virtual-reality-controller-1324150>
28. STATT, N. Samsung's Rink virtual reality controllers are janky, but the Gear VR needs them. In: *The Verge* [online]. 7. ledna. 2016 [cit. 2017-ledna-05]. Dostupné z: <http://www.theverge.com/2016/1/7/10727852/samsung-rink-motion-controllers-gear-vr-ces-2016>
29. BYE, K. Tracking Your Hands Using Flex Sensor Technology with Manus VR. In: *Road to VR* [online]. 10. září. 2016 [cit. 2017-ledna-05]. Dostupné z: <http://www.roadtovr.com/tracking-hands-using-flex-sensor-technology-manus-vr/>
30. TechTalks, Inside the Manus VR Development Kit. *Manus VR* [online]. [cit. 2017-ledna-05]. Dostupné z: <https://manus-vr.com>
31. LANG, B. Hands-on: Gloveone's Newest Haptic Glove Has Impressive Tracking Tech. In: *Road to VR* [online]. 15. června. 2016 [cit. 2017-ledna-05]. Dostupné z: <http://www.roadtovr.com/hands-on-gloveones-newest-haptic-glove-has-impressive-tracking-tech/>
32. JAMES, P. Senso is a VR Input Glove With Per-finger Haptics and Simulated Temperature. In: *Road to VR* [online]. 26. prosince. 2016 [cit. 2017-ledna-06]. Dostupné z: <http://www.roadtovr.com/senso-vr-input-glove-per-finger-haptics-simulated-temperature/>
33. LANG, B. Hands-on: Gloveone's Newest Haptic Glove Has Impressive Tracking Tech. In: *Road to VR* [online]. 15. červen. 2016 [cit. 2017-ledna-05]. Dostupné z: <http://www.roadtovr.com/hands-on-gloveones-newest-haptic-glove-has-impressive-tracking-tech/>
34. HAYDEN, S. PowerClaw is a Haptic Glove Ready To Freeze, Burn, and Shock You (Virtually). In: *Road to VR* [online]. 19. srpna. 2016 [cit. 2017-ledna-05]. Dostupné z: <http://www.roadtovr.com/powerclaw-haptic-glove-ready-freeze-burn-shock-virtually/>
35. ABRASH, M. Latency – the sine qua non of AR and VR. In: *ValveSoftware* [online]. 29. prosince. 2012 [cit. 2017-ledna-06]. Dostupné z: <http://blogs.valvesoftware.com/abrash/latency-the-sine-qua-non-of-ar-and-vr/>
36. MVN Products. *Xsens* [online]. [cit. 2017-ledna-06]. Dostupné z: <https://www.xsens.com/products/xsens-mvn/>

37. Perception Neuron. *Neuron MoCap* [online]. [cit. 2017-ledna-06]. Dostupné z: https://neuronmocap.com/products/perception_neuron
38. PrioVR. *Yostlabs* [online]. [cit. 2017-ledna-06]. Dostupné z: <https://yostlabs.com/priovr/>
39. FAQ. *VicoVR* [online]. [cit. 2017-ledna-06]. Dostupné z: <http://vicovr.com/faq>
40. JAMES, P. 'Teslasuit' Kickstarter is Live, Full Body Haptic Suits Start at £1199. In: *Road to VR* [online]. 4. leden. 2016 [cit. 2017-ledna-06]. Dostupné z: <http://www.roadtovr.com/teslasuit-kickstarter-is-live-full-body-haptic-suits-start-at-1799/>
41. JAMES, P. 'Teslasuit' is a Tactile Skin that Lets You Feel VR, Kickstarter Jan 1st. In: *Road to VR* [online]. 31. prosinec. 2015 [cit. 2017-ledna-06]. Dostupné z: <http://www.roadtovr.com/teslasuit-is-a-tactile-skin-that-lets-you-feel-vr-kickstarter-jan-1st/>
42. Atomic A3. *Atomic Motion Systems* [online]. [cit. 2017-ledna-06]. Dostupné z: http://www.atomicmotionsystems.com/?page_id=10244
43. Store. *Roto VR* [online]. [cit. 2017-ledna-06]. Dostupné z: <http://www.rotovr.com/store>
44. *VRGO* [online]. [cit. 2017-ledna-06]. Dostupné z: <http://www.vrgochair.com>
45. MiniPCs: VRGO. *Zotac* [online]. [cit. 2017-ledna-06]. Dostupné z: https://www.zotac.com/us/product/mini_pcs/vr-go
46. Virtuix Omni Package. *Virtuix* [online]. [cit. 2017-ledna-06]. Dostupné z: <http://www.virtuix.com/product/omni-package/>
47. Product. *Cyberith* [online]. [cit. 2017-ledna-06]. Dostupné z: <http://cyberith.com/product/>
48. GOODNER, S. Kat Walk omnidirectional VR treadmill lets you walk, jump, and sit freely. In: *New Atlas* [online]. 13. srpna. 2015 [cit. 2017-ledna-06]. Dostupné z: <http://newatlas.com/kat-walk-omnidirectional-vr-treadmill/38901/>
49. BOXALL, A. FIRST IMPRESSIONS: WIZDISH ROVR. In: *Digital Trends* [online]. 15. února. 2016 [cit. 2017-ledna-06]. Dostupné z: <http://www.digitaltrends.com/virtual-reality/wizdish-rovr-vr-shoes-first-impressions/>
50. JAMES, P. Hands on With the Latest Infinadeck Treadmill at CES 2016. In: *Road to VR* [online]. 7. ledna. 2016 [cit. 2017-ledna-06]. Dostupné z: <http://www.roadtovr.com/hands-on-with-the-latest-infinadeck-treadmill-at-ces-2016/>
51. WATT, A. *3D Computer Graphics*. třetí vydání. Addison-Wesley, 2000. ISBN 978-0-201-39855-7.
52. TIŠNOVSKÝ, P. Implicitní plochy (metaballs) v POV-Rayi. In: *Root.cz* [online]. 8. dubna. 2008 [cit. 2016-prosinec-28]. Dostupné z: <https://www.root.cz/clanky/implicitni-plochy-metaballs-v-pov-rayi/>
53. BERNARDINI, F. a RUSHMEIER, H. The 3D Model Acquisition Pipeline. *Computer Graphics Forum*. Blackwell Publishers Ltd, 2002, **21** (2), 149-72. ISSN 1467-8659.
54. HAVEMANN, S. *Generative Mesh Modeling*. Braunschweig (Německo): 2005. Diplomová práce. University of Technology, Institute of Computer Graphics.
55. 3D Studio Max Subscription. *Autodesk* [online]. [cit. 2016-prosinec-29]. Dostupné z: <http://www.autodesk.com/products/3ds-max/subscribe>
56. Maya vs. Maya LT. *Autodesk* [online]. [cit. 2016-prosinec-29]. Dostupné z: <http://www.autodesk.com/products/maya/compare/compare-products>
57. MASTERS, M. 3DS Max vs. Maya. In: *Digital-Tutors* [online]. 2014 [cit. 2016-prosinec-29]. Dostupné z: <http://blog.digitaltutors.com/3ds-max-vs-maya-is-one-better-than-the-other/>

58. MASTERS, M. 3DS Max, Maya LT or Blender. In: *Digital-Tutors* [online]. 2014 [cit. 2016-prosince-29]. Dostupné z: <http://blog.digitaltutors.com/3ds-max-maya-lt-blender-3d-software-choose-asset-creation/>
59. VisCube™ C2-4K, C4-4K: CAVE Immersive 3D Display. *VisBox* [online]. [cit. 2017-ledna-06]. Dostupné z: <http://www.visbox.com/products/cave/viscube-c4-4k/>

Obsah DVD

- Build aplikace
Složka obsahující spustitelný soubor aplikace
- Modely a textury
Složka obsahující modely, textury a animace v upravitelném formátu
- Text diplomové práce
Text této práce ve formátech pdf a docx
- Unity projekt
Složka obsahující aplikaci jako upravitelný Unity projekt
- Demonstrační video – DP Marek Zouhar.mp4
Video demonstrující vytvořenou aplikaci
- Návod.txt
Stručný návod pro ovládání spustitelné aplikace
- Obsah DVD.txt
Textový soubor obsahující shrnutí obsahu DVD

Manuál

Spuštění

Pro spuštění aplikace je nutné mít do počítače splňujícího hardwarové i softwarové požadavky zapojen HTC Vive a mít softwarově zkalibrovanou místnost. Aplikaci lze po zkopírování na disk spustit pomocí souboru "xzouha10DP.exe", který se spolu s daty potřebnými pro běh aplikace nachází na DVD ve složce "Build aplikace".

Ovládání aplikace

- **Kruhové menu**

Dotykem palce touchpadů ovladačů se zobrazí kruhové menu. Menu má dvě úrovně, kde v první se nachází kategorie (zleva) chladných, střelných a vrhacích zbraní a spodní kvadrant hlavního menu představuje teleport. Ten, případně kategorii předmětů, lze vybrat stiskem touchpadu v okamžik, kdy je požadovaná část zvýrazněna, k čemuž dojde, když je jí palec na touchpadu nejbližší. Každá kategorie obsahuje několik předmětů, které je možné vybrat stejným způsobem a rudé tlačítko sloužící pro zobrazení hlavního menu.

- **Teleport**

Po jeho výběru z hlavního menu je možné teleport použít pomocí tlačítka spouště ovladače. Po jeho stisku se zobrazí křivka, na jejíž konec je uživatel po uvolnění spouště přesunut.

- **Chladné zbraně**

Chladné zbraně nemají žádné speciální ovládání. Se světem interagují pouze pomocí úderů.

- **Střelné zbraně**

Zbraně této kategorie vystřelí po stisku spouště ovladače. Kdykoli je možné zbraň přebít prostřednictvím stisku tlačítka grip (dvě tlačítka umístěná na obou stranách ovladače takřka v dlani uživatele). Pistole dokáže přepnout z automatického režimu střelby na poloautomatický a zpět stiskem aplikačního tlačítka nad touchpadem.

- **Vrhací zbraně**

Pro hod musí uživatel stisknout spoušť ovladače. Při jejím uvolnění je drženému objektu předán vektor rychlosti ovladače a tak pokračuje v uživatelem definované trase. Granáty jsou stiskem spouště odjištěny a po třech vteřinách explodují.

- **Spawnergy**

Po mapě jsou rozprostřeny objekty, které po výstřelu nebo úderu do nich vytvoří jeden či více objektů ovlivněných fyzikou. Tvořící objekty mají podobu duté krychle nebo kanónu se znázorněním objektu, který vytváří, přičemž kanóny udělí vytvořenému objektu znatelnou rychlost.

- **Roboti**

Na mapě se nachází krácející a stojící roboti a spawnergy, které je tvoří. Každého robota je možné rozložit na části pomocí střelby, explozí či úderů.

- **Helikoptéry**

Nad mapou se vznášejí tři helikoptéry, přičemž aktivací (střelbou či úderem) každé z nich je zpomalen herní čas na třetinu. Opětovnou aktivací kterékoli z nich je čas vrácen na původní rychlost.

- **Restart**

Uživatel má možnost scénu kdykoli kompletně restartovat pomocí současného stisku všech tří hlavních vstupů ovladače: spouště, touchpadu a gripu.

Příloha 1







